Mobile Robot Navigation in Static and Dynamic Environments using Various Soft Computing Techniques

Anish Pandey

Department of Mechanical Engineering
National Institute of Technology Rourkela

Mobile Robot Navigation in Static and Dynamic Environments using Various Soft Computing Techniques

Dissertation submitted to the

National Institute of Technology Rourkela

in partial fulfillment of the requirements

of the degree of

Doctor of Philosophy

in

Mechanical Engineering

by

Anish Pandey

(Roll Number: 512ME119)

under the supervision of

Prof. Dayal R. Parhi

July, 2016

Department of Mechanical Engineering

National Institute of Technology Rourkela

Mechanical Engineering
National Institute of Technology Rourkela

July 18, 2016

Certificate of Examination

Roll Number: 512ME119

Name: Anish Pandey

Title of Dissertation: Mobile Robot Navigation in Static and Dynamic Environments using Various Soft Computing Techniques

We the below signed, after checking the dissertation mentioned above and the official record book (s) of the student, hereby state our approval of the dissertation submitted in partial fulfillment of the requirement of the degree of Doctor of Philosophy in Mechanical Engineering at National Institute of Technology, Rourkela. We are satisfied with the volume, quality, correctness, and originality of the work.

Dayal R. Parhi
Principal Supervisor

Bidyadhar Subudhi
Member (DSC)

Tarapada Roy
Member (DSC)

Sarat Kumar Das
Member (DSC)

Santosha Kumar Dwivedy
Examiner

Kalipada Maity
Chairman (DSC)

Mechanical Engineering
National Institute of Technology Rourkela

Dr. Dayal R. Parhi
Professor

July 18, 2016

Supervisor's Certificate

This is to certify that the work presented in this dissertation entitled *"Mobile Robot Navigation in Static and Dynamic Environments using Various Soft Computing Techniques"* by *"Anish Pandey"*, Roll Number: 512ME119, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Doctor of philosophy* in *Mechanical Engineering*. Neither this dissertation nor any part of it has been submitted for any degree or diploma to any institute or university in India or abroad.

Dayal R. Parhi

To my Parents,
with all my love

Declaration of Originality

I, Anish Pandey, Roll Number: 512ME119 hereby declare that this dissertation entitled *"Mobile Robot Navigation in Static and Dynamic Environments using Various Soft Computing Techniques"* represents my original work carried out as a doctoral student of NIT Rourkela and, to the best of my knowledge, it contains no material previously published or written by another person, nor any material presented for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the section "Bibliography". I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of my non-compliance detected in the future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

July 18, 2016

NIT Rourkela

Anish Pandey

Acknowledgment

My first thank is to the Almighty God, without whose blessings, I wouldn't have been writing this “acknowledgments".

I would like to extend my heartfelt indebtedness and gratitude to Prof. Dayal R. Parhi for his kindness in providing me an opportunity to work under his supervision and guidance. During this period, without his endless efforts, immense knowledge, deep patience, invaluable guidance and answers to my numerous questions, this research would have never been possible. I am especially obliged to him for teaching me both research and writing skills, which have been proven beneficial for my current research and future career. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. It has been a great honor and pleasure for me to do research under the supervision of Dr. Dayal R. Parhi. I am thankful to Prof. Sunil Kumar Sarangi, Director of National Institute of Technology, for giving me an opportunity to be a part of this institute of national importance and to work under the supervision of Prof. Dayal R. Parhi. I am sincerely obliged to Prof. S. S. Mahapatra, Head of the Department, Department of Mechanical Engineering, for providing me all official and laboratory facilities during the research period. His incessant encouragement towards research work has inspired me a lot.

I express my gratitude to Prof. P. K. Ray, Chairman DSC and DSC members for their indebted help and valuable suggestions for the accomplishment of the dissertation. I thank all the members of the Department of Mechanical Engineering, and the Institute, who helped me in various ways towards the completion of my work.

I would like to thank all my friends and lab-mates for their encouragement and understanding. Their support and lots of lovely memories with them can never be captured in words. Finally, I thank my parents, beloved wife, my little baby Atharv, and the entire family members for their unlimited support and strength.

July 18, 2016 — *Anish Pandey*

NIT Rourkela — Roll Number: 512ME119

Abstract

The applications of the autonomous mobile robot in many fields such as industry, space, defence and transportation, and other social sectors are growing day by day. The mobile robot performs many tasks such as rescue operation, patrolling, disaster relief, planetary exploration, and material handling, etc. Therefore, an intelligent mobile robot is required that could travel autonomously in various static and dynamic environments. The present research focuses on the design and implementation of the intelligent navigation algorithms, which is capable of navigating a mobile robot autonomously in static as well as dynamic environments.

Navigation and obstacle avoidance are one of the most important tasks for any mobile robots. The primary objective of this research work is to improve the navigation accuracy and efficiency of the mobile robot using various soft computing techniques. In this research work, Hybrid Fuzzy (H-Fuzzy) architecture, Cascade Neuro-Fuzzy (CN-Fuzzy) architecture, Fuzzy-Simulated Annealing (Fuzzy-SA) algorithm, Wind Driven Optimization (WDO) algorithm, and Fuzzy-Wind Driven Optimization (Fuzzy-WDO) algorithm have been designed and implemented to solve the navigation problems of a mobile robot in different static and dynamic environments. The performances of these proposed techniques are demonstrated through computer simulations using MATLAB software and implemented in real time by using experimental mobile robots. Furthermore, the performances of Wind Driven Optimization algorithm and Fuzzy-Wind Driven Optimization algorithm are found to be most efficient (in terms of path length and navigation time) as compared to rest of the techniques, which verifies the effectiveness and efficiency of these newly built techniques for mobile robot navigation. The results obtained from the proposed techniques are compared with other developed techniques such as Fuzzy Logics, Genetic algorithm (GA), Neural Network, and Particle Swarm Optimization (PSO) algorithm, etc. to prove the authenticity of the proposed developed techniques.

Keywords: Intelligent Mobile Robot; Navigation; Hybrid Fuzzy; Cascade Neuro-Fuzzy; Simulated Annealing algorithm; Wind Driven Optimization algorithm.

Contents

List of Figures

List of Tables

Symbols & Abbreviations

V_R	Right Wheel Linear Velocity
V_L	Left Wheel Linear Velocity
ω_R	Angular Velocity of Right Wheel
ω_L	Angular Velocity of Left Wheel
θ	Steering Angle (Turning Angle)
C	Center of Mass of a Mobile Robot
R	Radius of Wheel
V	Centre Linear Velocity of the Robot
ω	Centre Angular (Rotational) Velocity of Left Wheel
L	Track Width of the Robot
m	Total Mass of the Mobile Robot
I	Moment of Inertia of the Robot
τ_R	Right Wheel (Motor) Torques
τ_L	Left Wheel (Motor) Torques
d_f	Forward Obstacle Distance
d_l	Left Forward Obstacle Distance
d_r	Right Forward Obstacle Distance
m_r	Right Motor Velocity
m_l	Left Motor Velocity
TFa	Takagi-Sugeno Type Fuzzy Logic Architecture
MFa	Mamdani-Type Fuzzy Logic Architecture
F.O.D.	Front Obstacle Distance

L.O.D.	Left Obstacle Distance
R.O.D.	Right Obstacle Distance
T.A.	Turning Angle
RMV	Right Motor Velocity
LMV	Left Motor Velocity
FLA	Fuzzy logic architecture
SAA	Simulated Annealing Algorithm
WDO	Wind Driven Optimization
GA	Genetic Algorithm
PSO	Particle Swarm Optimization Algorithm
ACO	Ant Colony Optimization Algorithm
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network

Note: - The symbols and abbreviations other than above have been explained in the text.

Chapter 1

Introduction

1.1 Background and Motivations

Mobile robot is an autonomous agent capable of navigating intelligently anywhere using sensor-actuator control techniques. The mobile robot performs various tasks such as material handling in the industries, planetary exploration in the Mars and other planets, and other social sectors without human intervention. Current research in the field of mobile robotics focuses on designing and developing an intelligent algorithm or technique, which can control the motion and orientation of the mobile robot with obstacle avoidance/wall following competence in the static and dynamic environments. Successful autonomous mobile robot navigation in the environment depends on its technique/controller. Basically, during navigation, the mobile robot faces two types of obstacles: static and dynamic. Several techniques have been applied by the various researchers for mobile robot navigation and obstacle avoidance. According to literature survey, it is found that the static obstacle avoidance is comparatively easy from the dynamic obstacle avoidance. Therefore, the author is motivated to solve the static and dynamic obstacle avoidance problem using various soft computing techniques such as Hybrid Fuzzy (H-Fuzzy) architecture, Cascade Neuro-Fuzzy (CN-Fuzzy) architecture, Fuzzy-Simulated Annealing (Fuzzy-SA) algorithm, Wind Driven Optimization (WDO) algorithm, and Fuzzy-Wind Driven Optimization (Fuzzy-WDO) algorithm. The rest of this chapter is organized as follows: Section 1.2 introduces the aims and objectives of the proposed research work. Section 1.3 describes the methodologies applied for proposed research work. Section 1.4 presents the novelty of the proposed research work. Finally, Section 1.5 gives an outline of each chapter of the dissertation.

1.2 Aims and Objectives of the Proposed Research Work

The aims and objectives for the research towards mobile robot navigation in the static and dynamic environments are summarized below: -

- To analyze various soft computing techniques (such as H-Fuzzy architecture, CN-Fuzzy architecture, Fuzzy-SA algorithm, WDO algorithm, and Fuzzy-WDO) for navigating a mobile robot from start position to goal position while avoiding static and dynamic obstacles present in the environment.

- Integration of various sensors such as ultrasonic range finder sensors, sharp infrared range sensors for mapping the environment cluttered with dynamic and static obstacles.

- To design a simulated environment for carrying out the simulation exercises using the above mentioned soft computing techniques.

- To develop experimental setup to perform the experimental exercises using the above mentioned soft computing techniques.

1.3 Methodologies Applied for Proposed Research Work

In this research work, Hybrid Fuzzy (H-Fuzzy) architecture, Cascade Neuro-Fuzzy (CN-Fuzzy) architecture, Fuzzy-Simulated Annealing (Fuzzy-SA) algorithm, Wind Driven Optimization (WDO) algorithm, and Fuzzy-Wind Driven Optimization (Fuzzy-WDO) algorithm have been designed and implemented to solve the navigation problems of a mobile robot in different environments.

The methodologies applied for proposed research work is summarized as follows: -

- To study the various techniques applied to the mobile robot navigation in the literature survey.
- To study the kinematic and dynamic analysis of the nonholonomic differential drive wheeled mobile robot.
- To develop the Hybrid Fuzzy (H-Fuzzy) architecture for intelligent mobile robot navigation and obstacle avoidance in the static and dynamic environments.

- To design a Cascade Neuro-Fuzzy (CN-Fuzzy) architecture to improve the navigation and obstacle avoidance strategies of the mobile robot in various (static and dynamic) environments.
- To integrate the Takagi-Sugeno fuzzy model with the simulated annealing algorithm called as Fuzzy-Simulated Annealing (Fuzzy-SA) algorithm to optimize the navigation path length of the mobile robot in the given environment.
- To apply a Wind Driven Optimization (WDO) algorithm to solve the optimal path planning problems of a mobile robot in various simulation and experimental environments.
- To make a hybridization of the Fuzzy-Wind Driven Optimization algorithm to adjust and tune the input/output membership function parameters of the fuzzy controller. This developed algorithm improves the navigation performance of the mobile robot in the given environments and produces a smooth navigation path within a reasonable time.
- To make a comparative study of all proposed developed techniques for checking its strength and weakness in the various environments.
- To demonstrate the various simulation and experimental results of the proposed techniques using the simulation and experimental setup.

1.4 Novelty of the Proposed Research Work

In literature survey, it is found that most of the researchers have applied the various soft computing techniques for mobile robot navigation in only static environments. However, few researchers have considered dynamic environments for mobile robot navigation. The novelty of this dissertation is to design, analysis, and develop soft computing techniques such as H-Fuzzy architecture, CN-Fuzzy architecture, Fuzzy-SA algorithm, WDO algorithm, and Fuzzy-WDO algorithm for mobile robot navigation and obstacle avoidance in the static as well as dynamic environments.

In this research work, the application of Wind Driven Optimization (WDO) algorithm for the mobile robot navigation has been carried out. Besides, this WDO algorithm is integrated with the fuzzy controller to adjust and optimize the antecedent and consequent parameters of the fuzzy membership function and is not found during the literature survey.

1.5 Outline of the dissertation

The rest of this dissertation is organized below: -

- **Chapter-2** introduces the literature review of the kinematic and dynamic analysis of wheeled mobile robot, and various soft computing techniques applied for mobile robot navigation.
- **Chapter-3** demonstrates the kinematic and dynamic analysis of nonholonomic differential drive wheeled mobile robot.
- **Chapter-4** presents the intelligent navigation of mobile robot in the various static and dynamic environments using Hybrid Fuzzy (H-Fuzzy) Architecture.
- **Chapter-5** describes the intelligent navigation control of mobile robot in the various (static and dynamic) environments using Cascade Neuro-Fuzzy (CN-Fuzzy) Architecture.
- **Chapter-6** presents the mobile robot navigation among the stationary and moving obstacle in the environments using Takagi-Sugeno Fuzzy Model and Simulated Annealing (Fuzzy-SA) Algorithm Controller.
- **Chapter-7** introduces the optimum navigation of mobile robot in the simulation and experimental environments using Wind Driven Optimization (WDO) Algorithm.
- **Chapter-8** introduces the optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization (Fuzzy-WDO) Algorithm.
- **Chapter-9** presents the comparative study of all the proposed soft computing techniques applied for mobile robot navigation.
- Finally, **Chapter-10** describes the conclusion and scope for future research.

Chapter 2

Literature Review

2.1 Introduction

This chapter introduces the literature survey of the various techniques used for mobile robot navigation. Navigation and obstacle avoidance are one of the fundamental problems in mobile robotics, which are being solved by the various researchers in the past two decades. The aim of navigation is to search an optimal or suboptimal path from the start point to the goal point with obstacle avoidance competence [1]. Basically, the mobile robot navigation has been done by the Deterministic algorithm and Nondeterministic (Stochastic) algorithm. Nowadays, the hybridization of both the algorithms called as an Evolutionary algorithm is being used to solve the mobile robot navigation problem. Figure 2.1 shows the general classification of the Deterministic algorithm, Nondeterministic (Stochastic) algorithm, and Evolutionary algorithm, which are implemented for mobile robot navigation by various authors.

Navigation is an essential task in the field of mobile robotics, which can be classified into two types: global navigation and local navigation. In the global navigation, the prior knowledge of the environment should be available. Many methods have been developed for global navigation, i.e. Voronoi graph [2, 3], Artificial potential field method [4, 5], Dijkstra algorithm [6], Visibility graph [7], Grids [8], and Cell decomposition method [9], and so on. In the local navigation, the robot can decide or control its motion and orientation autonomously using equipped sensors such as ultrasonic range finder sensors, sharp infrared range sensors, and vision (camera) sensors, etc. Fuzzy logic [10], Neural network [11], Neuro-fuzzy [12], Genetic algorithm [13], Particle swarm optimization algorithm [14], Ant colony optimization algorithm [15], and Simulated annealing algorithm [16], etc. are successfully employed by various researchers to solve the local navigation problem.

Rest of the chapter is organized as follows: Section 2.2 presents the literature survey of kinematic and dynamic analysis of the wheeled mobile robots. Section 2.3 discusses the literature review of various soft computing techniques used for mobile robot navigation. Finally, Section 2.4 describes the summary of this literature survey.

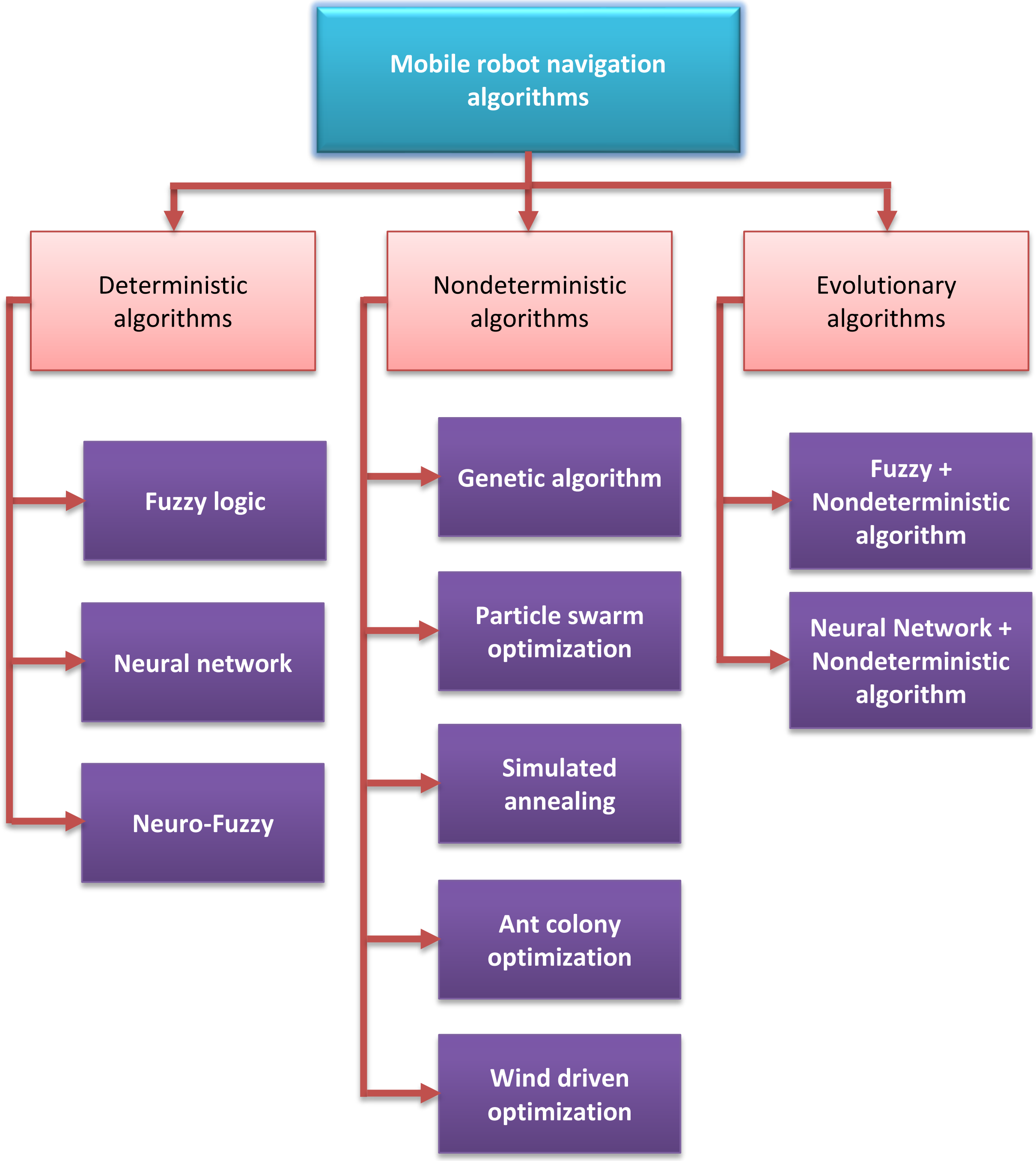


Figure 2.1: General classification of the Deterministic algorithm, Nondeterministic (Stochastic) algorithm, and Evolutionary algorithm used for mobile robot navigation.

2.2 Kinematic and Dynamic Analysis of the Wheeled Mobile Robot

The motion control problem of an autonomous wheeled mobile robot has been widely investigated in past decades. In recent years, there has been a growing interest in the design and development of an autonomous wheeled mobile robot using various soft computing techniques. In [17], the authors have studied the kinematic and dynamic constraints of a car-like mobile robot and applied it to navigation among moving obstacles in the environments using neuro-fuzzy approaches. Abadi and Khooban [18] have solved the trajectory tracking problem of nonholonomic wheeled mobile robots using Random Inertia Weight Particle Swarm Optimization (RNW-PSO) based optimal Mamdani-type fuzzy controller. The motion problem of the wheeled mobile robots on uneven terrain has been addressed in [19]. Wang and Yang [20] have developed the neuro-fuzzy controller for navigation of a nonholonomic differential drive mobile robot (shown in Figure 2.2). The combination of four sharp infrared sensors is equipped on the robot to read the obstacle distance, and this distance information is fed to the controller to adjust the speed of two separate motors of the robot.

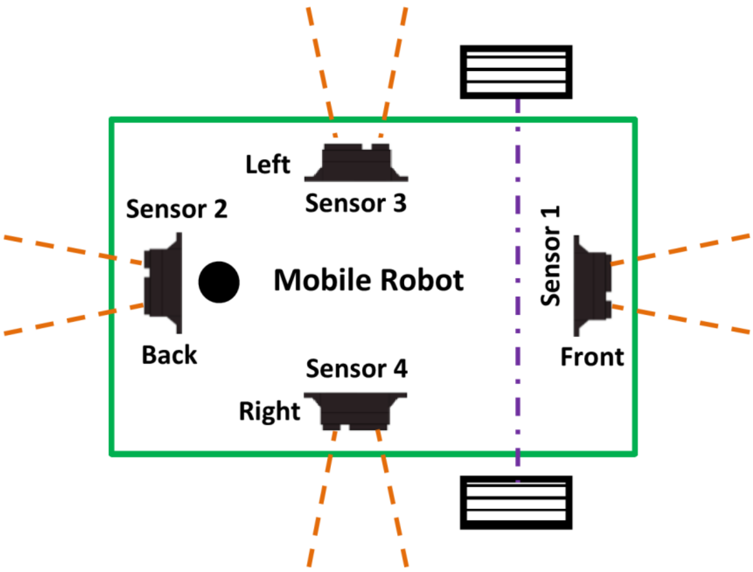


Figure 2.2: Infrared sensor based nonholonomic differential drive mobile robot developed by Wang and Yang [20].

Wheeled mobile robots [21] have been widely used in various industrial applications, transportation, and social sectors, etc. Martinez et al. [22] have designed the kinematics and dynamics trajectory tracking control of the autonomous unicycle mobile robot using type-2 fuzzy logic and genetic algorithms. An adaptive neural network based motion and orientation control of a nonholonomic wheeled mobile robot has been presented in [23]. Liang et al. [24] have presented the kinematic modelling of the two-wheeled differential drive mobile robot.

2.3 Various Soft Computing Techniques used for Mobile Robot Navigation

In the past few years, many soft computing techniques are proposed by the researchers to solve the robot navigation and obstacle avoidance problem in the various environments. The various soft computing techniques applied for mobile robot navigation in the different static and dynamic environments are summarized below.

2.3.1 Fuzzy Logic Technique for Mobile Robot Navigation

The concept of fuzzy logic has been introduced by Zadeh [25], which is extensively used in many engineering applications such as mobile robotics, image processing, etc. This method plays a vital role in the field of mobile robots. The fuzzy logic technique has been successfully applied by many researchers to control the position and orientation of mobile robot in the environment. Ren et al. [26] have designed an intelligent fuzzy logic controller to solve the navigation problem of wheeled mobile robot in an unknown and changing environment. Fuzzy logic systems are inspired by human reasoning, which works based on perception. In [27], the authors have presented the Gradient method based optimal Takagi-Sugeno fuzzy controller to tune the membership function parameters, and applied it to mobile robot navigation and obstacle avoidance. Qing-yong et al. [28] have presented the behavior-based fuzzy architecture for mobile robot navigation in unknown environments. They have designed four basic behaviors: goal-seeking behavior, obstacle avoidance behavior, tracking behavior, etc. for mobile robot navigation and tested it in various simulation environments. The eight rule-based fuzzy controllers have been designed by Boubertakh et al. [29] for obstacle avoidance and goal-seeking behavior of the mobile robot. Muthu et al. [30] have presented the Atmega microcontroller based fuzzy logic controller (Figure 2.3) for the wheeled mobile robot. The proposed controller train the mobile robot to navigate in an environment without any human intervention. The controller receives inputs (obstacle distance) from the group of sensors to control the right and left motor of the mobile robot.

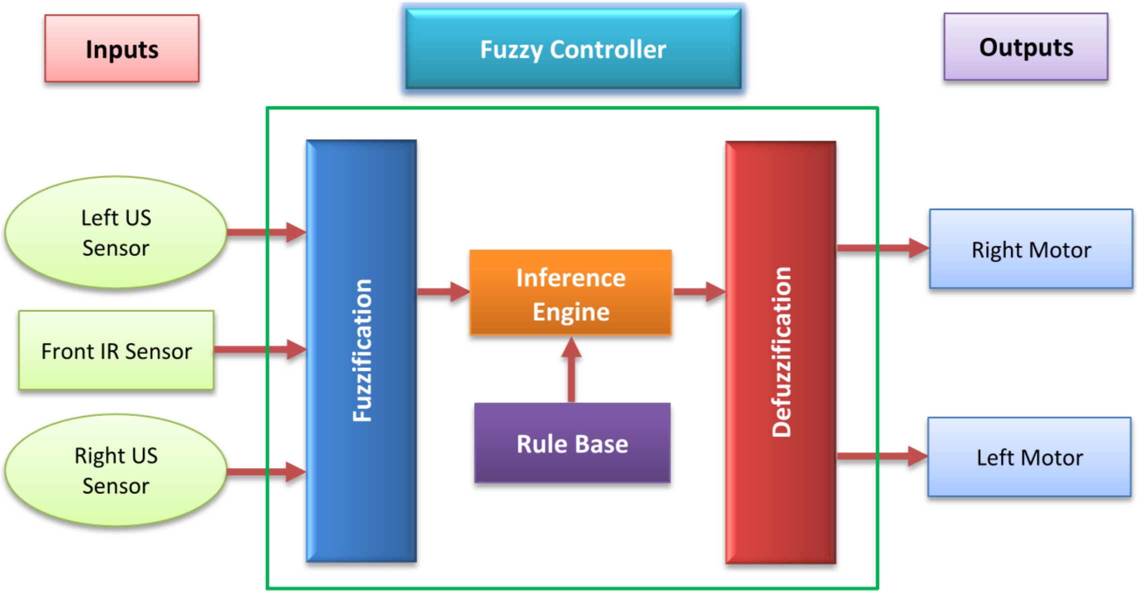


Figure 2.3: The block diagram of the fuzzy controller designed by Muthu et al. [30].

The sensor-based mobile robot navigation in an indoor environment using a fuzzy logic controller has been discussed in [31-32]. Wu et al. [33] have developed the sensor based mobile robot navigation in the narrow environment using fuzzy controller and genetic algorithm. Where the fuzzy controller provides the initial membership function and the genetic algorithm choose the best membership value to optimize the fuzzy controller for mobile robot navigation. Obstacle avoidance is very important for successful navigation of autonomous mobile robot. Samsudin et al. [34] have combined the reinforcement learning method and genetic algorithm to optimize the fuzzy controller for improving their performance when the mobile robot moves in an unknown environment. Fuzzy reinforcement learning sensor-based mobile robot navigation has been presented by Beom and Cho [35] for complex environments. Pradhan et al. [36] have used fuzzy logic controller with different membership functions for the navigation of one thousand robots in an entirely unknown environment. The authors have compared the performance of different membership functions such as triangular, trapezoidal and gaussian for mobile robot navigation and stated that the gaussian membership function is more efficient for navigation. In [37], the authors have combined the fuzzy genetic algorithm to solve the path planning and control problem of an autonomous mobile robot (AMR) using ultrasonic range finder sensor information. Farooq et al. [38] have presented the comparative study between the zero order Takagi-Sugeno and Mamdani-type fuzzy logic models for mobile robot navigation and obstacle avoidance. Both the controllers receive inputs (obstacle distance) from the left and right ultrasonic sensors to control the left and right velocities of the motors of the mobile robot. During comparison study, the authors have found that in terms of smoothness Mamdani-type fuzzy model gives a better result. On the other hand, the Takagi-Sugeno fuzzy model takes less memory space in the real-time microcontroller implementation.

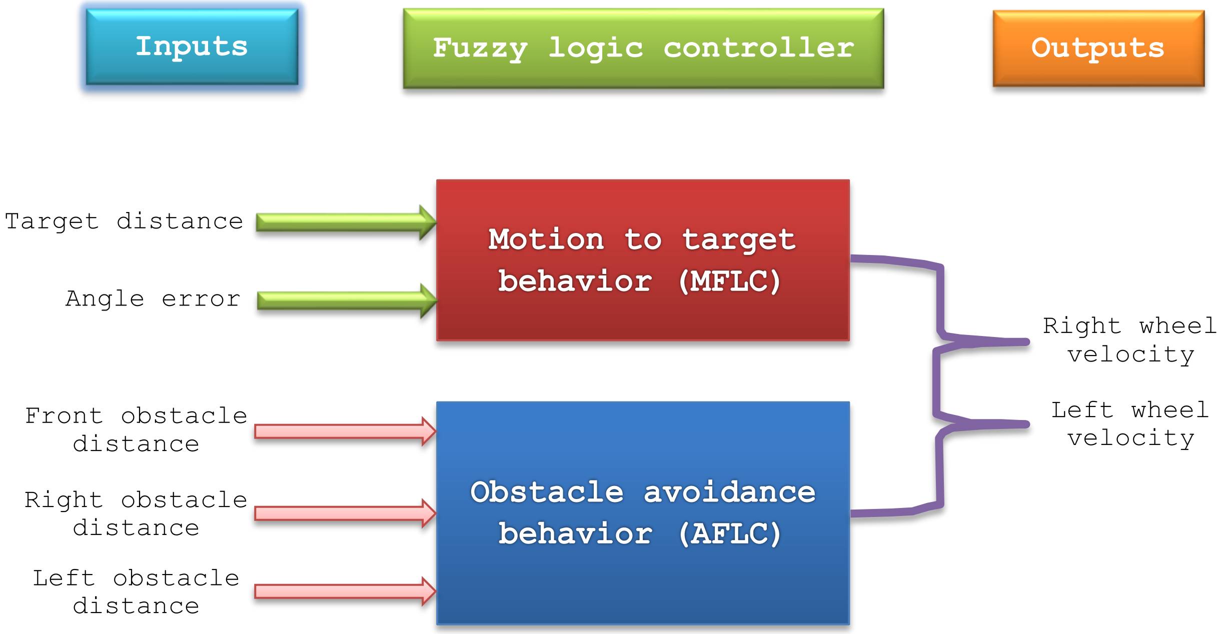


Figure 2.4: Behavior based fuzzy controller for mobile robot navigation and obstacle avoidance developed by Algabri et al. [39].

Hybridization of Fuzzy and Nondeterministic Algorithm

Algabri et al. [39] have combined the fuzzy logic with other soft computing techniques such as Genetic Algorithm (GA), Neural Networks (NN), and Particle Swarm Optimization (PSO) to optimize the membership function parameters of the fuzzy controller for improving the navigation performance of mobile robot. They have designed two basic fuzzy logic behaviors: Motion to target behavior (MFLC) and obstacle avoidance behavior (AFLC) as shown in Figure 2.4. In [40], the authors have developed genetic-fuzzy and genetic-neural for an adaptive navigation planning of a car-like mobile robot between dynamic obstacles. In this study, the genetic algorithm is employed to adjust the fuzzy membership function and weight of the neural network. Fuzzy PWM (Pulse Width Modulation) controller has been presented in the article [41] for mobile robot navigation and obstacle avoidance in an unknown environment. Abdessemed et al. [42] have designed an evolutionary algorithm to optimize the antecedent and consequent parameters of the fuzzy controller, and implemented it for mobile robot path planning. Selekwa et al. [43] have presented the fuzzy behavior controller for mobile robot navigation in the densely obstacle populated environments. The authors have designed two behavior control actions for navigation, namely obstacle avoidance behavior and the goal-seeking behavior. The obstacle avoidance behavior is done by range finding sensors, which detects the nearest obstacle distance, and the goal-seeking behavior is made by compass measurements, which determines the direction of the goal. Pratihar et al. [44] have developed a genetic-fuzzy technique based on a combined approach of genetic algorithm and fuzzy logic (GA-FL) to solve the mobile robot motion planning problems in the dynamic environments. Sensor-based wireless fuzzy controller has been designed by Faisal et al. [45] for mobile robot navigation in the industries among the static and dynamic objects. The two fuzzy controllers: tracking fuzzy logic control (TFLC) and obstacle avoidance fuzzy logic control (OAFLC) are helping the robot to search collision-free path from the start point to goal point. Babalou and Seifiour [46] have developed the sensor-based on-line path planning method for the mobile robot in dynamic environments. Li et al. [47] have designed the four types of fuzzy controller: wall-following fuzzy, corner control fuzzy, garage-parking fuzzy and parallel-parking fuzzy for the car-like mobile robot (CLMR). The developed fuzzy controllers have been implemented real-time using field-programmable gate array (FPGA) chip, and tested it in various experimental scenarios. Li and Chang [48] have presented a real-time fuzzy target tracking control scheme for autonomous mobile robots using infrared sensors. The behavior-based fuzzy logic controller has been made by Dongshu et al. [49] to solve the navigation problem of mobile robot in unknown dynamic environment. The different fuzzy rule-based controller has been constructed to deal with different behavior and also helps the robot to get out from the trapped situations. Antonelli et al. [50] have presented the path-following approach for differential drive mobile robots using the fuzzy logic technique. The designed fuzzy rules are able to emulate the human driving behavior. Ayari et al. [51] have developed a multi-agent fuzzy logic intelligent control system, which trains the robot to navigate autonomously in dynamic and uncertain environments.

2.3.2 Neural Network Technique for Mobile Robot Navigation

The neural network is one of the important technique for the mobile robot navigation. This neural network technique is motivated from the human brain, which is being applied by many researchers in the different fields such as signal and image processing, pattern recognition, mobile robot path planning, and business, etc. Zou et al. [52] have presented the literature survey of neural networks and its applications in mobile robotics. In [53], the authors have combined the multi-layer feed forward artificial neural network with Q-reinforcement learning method to construct a robust path-planning algorithm for the mobile robot. Rai and Rai [54] have designed the Arduino Uno microcontroller-based DC motor speed control system using the Multilayer neural network controller and Proportional Integral Derivative (PID) controller. Patino and Carelli [55] have designed the automatic steering controller for a mobile vehicle using neural network architecture. Yang and Meng [56] have applied the biologically inspired neural network to generate a collision-free path in a nonstationary environment. Biologically inspired neural network based wall-following mobile robot has been presented by Nichols et al. [57]. Online path planning between unknown obstacles in the environment is an interesting problem in the field of mobile robotics. Motlagh et al. [58] have presented the target seeking, and obstacle avoidance behaviors using neural networks and reinforcement learning. Mobile robot navigation using hybrid neural network has been addressed by Gavrilov and Lee [59]. Singh and Parhi [60] have designed multilayer feed forward neural network (Figure

2.5), which controls the steering angle of the robot autonomously in the static and dynamic environments. The different obstacle distances are the inputs of the four-layered neural network, and the steering angle is the output. Real-time collision-free path planning becomes more difficult when the robot is moving in a dynamic and unstructured environment.

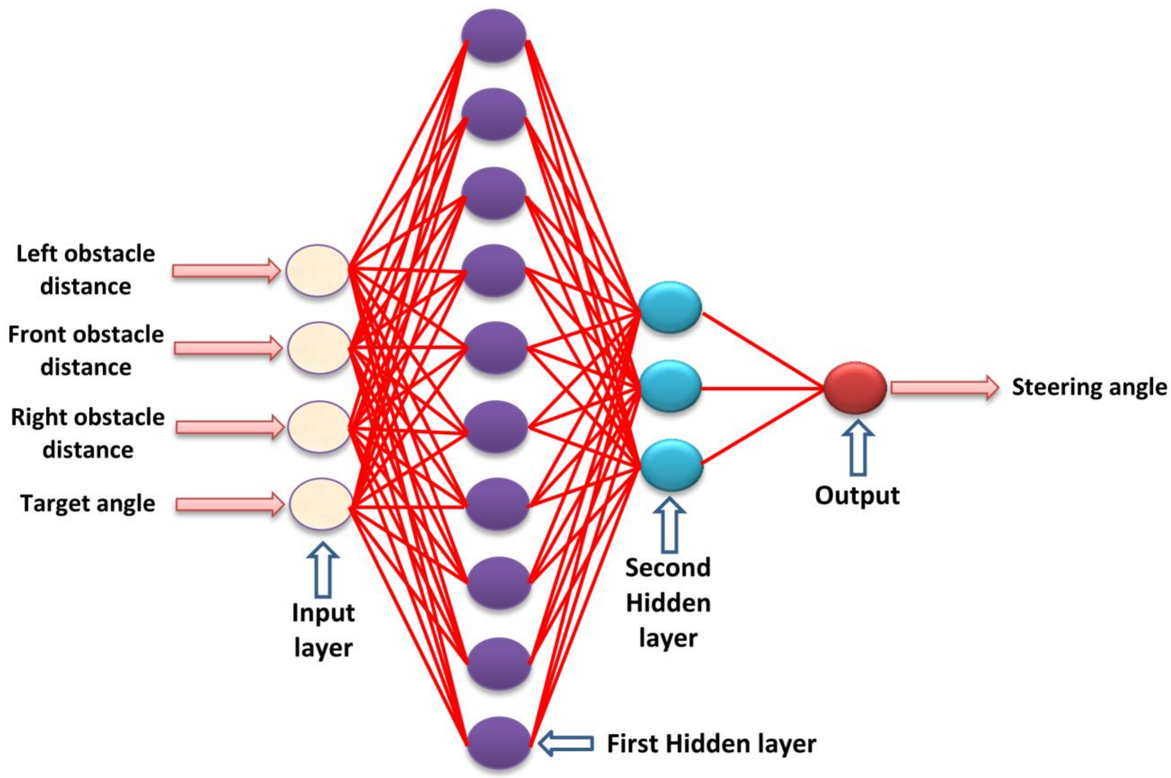


Figure 2.5: Four-layered neural network for mobile robot navigation designed by Singh and Parhi [60].

Hybridization of Neural Network and Nondeterministic Algorithm

Rossomando and Soria [61] have designed an adaptive neural network PID controller to solve the trajectory tracking control problem of a mobile robot. Al-Jarrah et al. [62] have described the path planning and coordination of multiple mobile robots using probabilistic neuro-fuzzy architecture. The authors have applied leader-followers concept to control their position and orientation in the working environment, where the follower robots behave like a leader robot. This proposed probabilistic neuro-fuzzy architecture is the combination of first order Sugeno fuzzy inference model and Adaptive Neuro-Fuzzy

Inference System (ANFIS). The fuzzy model has been used to control the linear and angular velocities of the leader robot and the follower robots, and ANFIS is implemented for automatic rule generation from the numerical dataset. In [63], the author has presented a neural network-based technique for intelligent path planning and control of a mobile robot. The two neural network controllers are applied to path planning and control. The first neural network controller helps the robot to search free space in the environment, and the second neural network controller trains the robot for obstacle avoidance. Glasius et al. [64] have used Hopfield neural network for path planning and obstacle avoidance in the complex environment. In [65], the authors have proposed type-2 fuzzy neural network (IT2FNN) to solve the obstacle avoidance and position stabilization problems of wheeled mobile robots. IT2FNN consists of three layers: input layer, hidden layers, and output layer. This proposed IT2FNN has four inputs: distance between the robot and goal point, distance between the robot and nearest obstacle, goal angle, and obstacle angle. The outputs of the IT2FNN are linear and angular velocities of the robot. Mahmud et al. [66] have presented the vision (camera) sensor based Kohonen-type artificial neural network for intelligent navigation of mobile robot. Chohra et al. [67] have designed intelligent autonomous navigation structure for a vehicle using multi-layered neural networks (NN). Brahmi et al. [68] have solved the path planning and localization problem of mobile robot using recurrent neural network (RNN). This RNN allows the robot to navigate autonomously in the unknown environments. In ref. [69], the authors have controlled the torque dynamic of nonholonomic mobile robot using neural network architecture.

2.3.3 Neuro-Fuzzy Technique for Mobile Robot Navigation

Zhu and Yang [12] have presented a neuro-fuzzy sensor based reactive navigation of mobile robots in unknown environments. Forty-eight Fuzzy rules and two behaviors, target seeking, and obstacle avoidance are designed using this model. A neural network based learning techniques is developed to tune the parameters of membership functions, which reduces the navigation path length from a start position to the end position in an environment. Al Mutib and Mattar [70] have proposed the sensor-based navigation of mobile robot using neuro-fuzzy architecture. The authors have used eight ultrasonic range finder sensors for surrounding obstacle detection as the input of the neuro-fuzzy controller for selecting the correct left and the right wheel speeds for a mobile robot.

Godjevac and Steele [71] have integrated the Takagi-Sugeno type fuzzy controller and Radial basis function neural network (RBFNN) to solve the mobile robot path planning. Where, the fuzzy logic is used to handle the uncertainty of the environment, and the neural network is used to tune the parameters of membership functions. In [72], the authors have constructed behaviour-based neuro-fuzzy control architecture (Figure 2.6) for a mobile robot navigation in an unstructured environment. The neural network is used to train the robot to reach the goal, and fuzzy architecture is integrated with it to control the velocities of the robot.

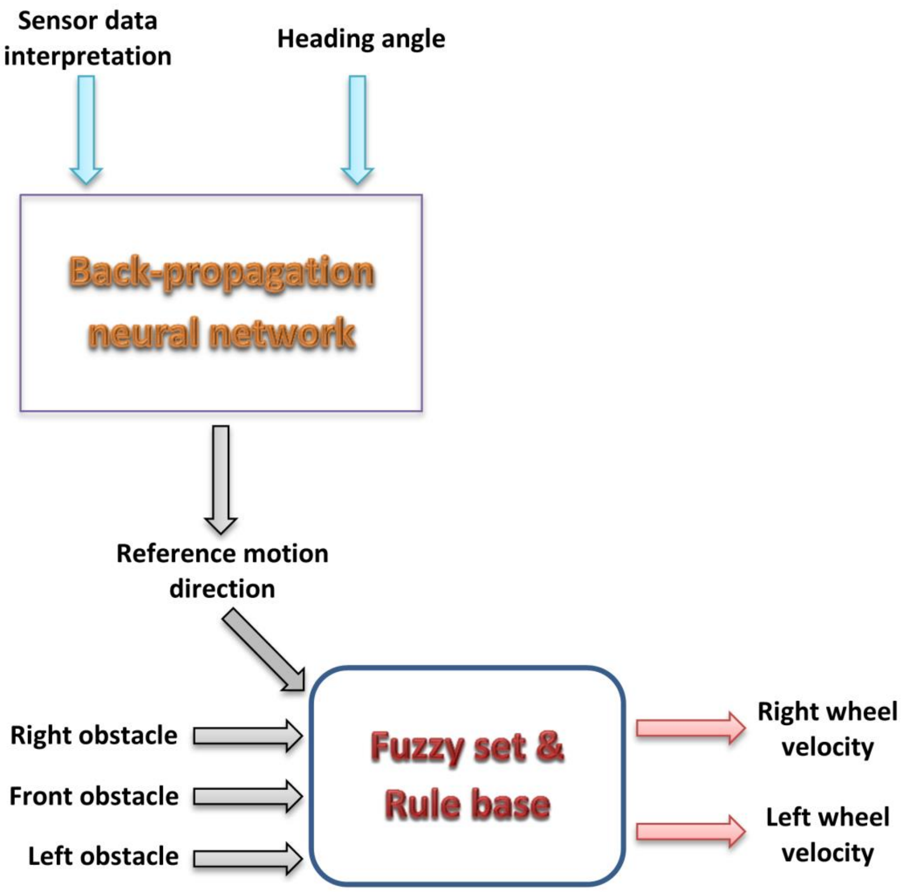


Figure 2.6: A neuro-fuzzy architecture for mobile robot navigation in uncertain environments developed by Li et al. [72].

Joshi and Zaveri [73] have developed a neuro-fuzzy system for reactive navigation and control of a mobile robot in the environment with the presence of static and dynamic obstacles. Marichal et al. [74] have designed a neuro-fuzzy sensor-actuator control technique to steer the mobile robot in unknown environments. RAM based neuro-fuzzy approach for mobile robot navigation has been presented by Zhang et al. [75]. They have used the fuzzy rule-based controller to interpret sensory information, and neural network controls the heading angle of the robot during navigation. Baturone et al. [76] have designed a low-cost embedded neuro-fuzzy controller for navigation of car-like mobile robot between the obstacles. Ma et al. [77] have used mixed soft computing techniques like fuzzy inference system and neural network to improve the learning and decision-making speed of a robot in unknown environments. Imen et al. [78] have applied the Adaptive Neuro-Fuzzy Inference System (ANFIS) technique to solve the path tracking problem of the nonholonomic wheeled mobile robots. They have used gradient descent learning algorithm to adjust the membership function parameters of the ANFIS. In [79], the authors have designed the two controllers: a Fuzzy Logic (FL) controller for obstacle avoidance and Artificial Neural Network (ANN) for wall-following of the mobile robot. Both the controllers receive inputs from the different sensors to avoid the obstacles when the robot moves towards the desired goal. Zhao and Wang [80] have incorporated sonar sensors with the neural network to solve the navigation problem of the autonomous mobile robot.

Kumar and Dhama et al. [81] have integrated the neural network and fuzzy logic to control the motion and orientation of the mobile robot in the crowded unknown environment. In their work, the authors have used fuzzy rule-based and neural network for goal reaching and actuator control, respectively. Song et al. [82] have designed a heuristic fuzzy-neuro network to create a mapping between the ultrasonic sensor data and velocity command of the robot. They have used sixteen rules to control the direction of the mobile robot. In [83], the authors have developed a Takagi-Sugeno type recurrent neuro fuzzy system and hybrid algorithm (genetic algorithm with particle swarm optimization) to improve the path tracking stability of the mobile robots. The neuro-fuzzy systems have been classified into two categories [84]: adaptive neuro-fuzzy systems (ANFIS) and hybrid neuro-fuzzy systems. Deshpande and Bhosale [84] have discussed the navigation of a nonholonomic wheeled mobile robot using ANFIS controller. Rusu and Petriu et al. [85] have presented a sensor-based neuro-fuzzy controller for mobile robot navigation in indoor environments. They have used infrared and contact sensors for target seeking and obstacle avoidance behavior. Pothal and Parhi [86] have proposed a sensor based adaptive neuro-fuzzy inference controller for navigation of single and multiple mobile robots in the highly cluttered environment. The authors have designed control architecture, which is able to avoid obstacle autonomously in various situations and reach the target efficiently. Neural network integrated fuzzy controller has been designed by Ng and Trivedi [87] for mobile robot navigation and wall-following control. In their work, the authors have used only five rules to control the steering angle, heading direction, and speed of the robot during wall-following. Demirli and Khoshnejad [88] have developed sensor-based neuro-fuzzy controller for autonomous parallel parking of a car-like mobile robot (CLMR). The proposed model received data from the sonar sensors to control the turning angle of CLMR. Al-Mayyahi et al. [89] have applied ANFIS technique for autonomous ground vehicle (AGV) navigation. In this work, they have designed four ANFIS controllers to control the left and right angular velocities, and angle between the robot and target (heading angle). In [90], the authors have designed a navigational approach for multiple mobile robots using a neuro-fuzzy controller. The proposed controller receives input (obstacle distance) from the array of sensors to actuate the left and right wheel velocities of the mobile robots. Algabri et al. [91] have applied ANFIS controller for mobile robot navigation and obstacle avoidance in an unknown environment. The authors have presented many simulation tests using Khepera Simulator (KiKs).

2.3.4 Genetic Algorithm for Mobile Robot Navigation

Ghorbani et al. [13] have solved the global path planning problem of a mobile robot in the complex environment using genetic algorithm approach. Elshamli et al. [92] have presented a genetic algorithm technique for solving the path planning problem of a mobile robot in static and dynamic environments. Mohanta et al. [93] have designed Petri-GA technique to optimize the navigation path length of multiple mobile robots in the cluttered environment. Kubota et al. [94] have used the fuzzy controller to guide the mobile robot in a static and dynamic environment, and the conventional genetic algorithms (GAs) are integrated with it, to optimize the navigation path length. Tuncer and Yildirim [95] have proposed a new mutation operator for a genetic algorithm (GA) and applied it for mobile robot navigation in the dynamic environments. Moreover, the authors have tested their developed method in various simulation environments and compared it with traditional GA techniques and stated that their developed mutation operator based GA performs better over traditional GA. In [96], the authors have designed a genetic algorithm to choose the best membership parameters from the fuzzy inference system and implemented it to control the steering angle of a mobile robot in the partially unknown environment. Hu et al. [97] have designed the knowledge-based genetic algorithm for mobile robot navigation between U-shaped obstacle and maze environment.

Liu et al. [98] have presented the optimal path planning technique for a mobile robot using fuzzy logic and genetic algorithm. The fuzzy controllers are applied to modify the moving direction of the mobile robot according to the obstacle distance received from the sensors, and genetic algorithm is used to adjust and tune membership function and rules. Improved genetic algorithm based mobile robot navigation has been proposed by the Li et al. [99]. The authors have done many simulation tests in the both static and dynamic environments to show the effectiveness of the proposed algorithm. Qu et al. [100] have developed the improved genetic algorithm instead of a conventional genetic algorithm for global path planning of the multiple mobile robots. The advantages of the improved genetic algorithm are capable of guiding the mobile robots efficiently from the starting node to end node without any collision in the environment. In [101], the authors have implemented Genetic-Fuzzy Controller (GA-FLC) to optimize and tune the Gaussian membership function parameters for mobile robot motion control. Castillo et al. [102] have designed Multiple Objective Genetic Algorithm (MOGA) for navigation path optimization of the mobile robot. Arora et al. [103] have presented the single fitness based genetic algorithm for solving the navigation problem in the dynamic environments. They have designed a fitness function based on the Euclidean distance formula between the robot and obstacle.

2.3.5 Simulated Annealing Algorithm for Mobile Robot Navigation

The concept of simulated annealing algorithm has come from statistical mechanics [104]. The simulated annealing is an iterative search algorithm inspired by the annealing of metals [105]. Miao and Tian [16] have applied the heuristic method based simulated annealing algorithm for robot path planning in the dynamic environments. The authors have compared this proposed algorithm to the Dijkstra algorithm and stated that the proposed algorithm consumes less processing time to get a solution compared to Dijkstra algorithm. Sensor-based autonomous navigation of a mobile robot in the dynamic environment has been presented by Chang and Song [106]. Martinez-Alfaro et al. [107] have developed the simulated annealing and fuzzy logic for designing an automatic path planning technique for mobile robot. The simulated annealing algorithm is used to search a collision-free optimal trajectory between the fixed polygonal obstacles, and forty-nine fuzzy rules are applied to adjust the velocity of the robot during navigation. Zhu et al. [108] have presented the global path planning method for a mobile robot using Artificial Potential Field (APF) method and Simulated Annealing Algorithm (SAA). In [109], the authors have used SAA with fuzzy logic to adjust and optimize the antecedent and the consequent parameters of the fuzzy membership function and applied it to solve the optimization problem of the servo systems. Janabi-Sharifi and Vinke [110] have addressed the local and global navigation problems in the real environment using Artificial Potential Field method and Simulated Annealing Algorithm. Tavares et al. [111] have discussed the off-line path planning problem of a mobile robot using SAA. They have designed some adaptive tuning parameters to change the behavior of that algorithm. Due to the slow convergence rate of the conventional simulated annealing algorithm, the Liang and Xu [112] have presented a modified simulated annealing algorithm, and applied it to mobile robot global path planning.

Nakamura and Kehtarnavaz [113] have designed an optimal fuzzy logic controller for autonomous mobile robot navigation and hurdle avoidance using a genetic algorithm and SAA combinatorial optimization techniques. Hussein et al. [114] have designed three metaheuristic optimization algorithms: Tabu Search, Simulated Annealing and Genetic Algorithm; and implemented these algorithms to improve the navigation performance of mobile robot from the start point to goal point in an environment. Miao and Tian [115] have presented a simulated annealing algorithm based intelligent navigational controller, which helps the robot to search an optimal or near-optimal path in the static and dynamic environments. Zhang et al. [116] have combined the simulated annealing algorithm and Ant Colony Optimization (ACO) algorithm to increase the navigation speed of the mobile robot. In [117], the authors have improved the convergence speed of the simulated annealing algorithm using the artificial neural network and applied it to mobile robot path planning. Synodinos and Aspragathos [118] have integrated simulated annealing algorithm and artificial potential field method to rescue the robot from undesired local minima problem during navigation. Zhao and Zu [119] have developed a Modified Particle Swarm Optimization (MPSO) technique for mobile robot navigation in the dynamic environment.

2.3.6 Particle Swarm Optimization Algorithm for Mobile Robot Navigation

Particle swarm optimization (PSO) is a population-based stochastic algorithm, which is inspired by the social behavior of bird flocks. PSO algorithm is used to find an optimal or near optimal solution of the problem using fitness function $f(x) = f(x_1, x_2, x_3, ...x_n)$, where x_i is a population of the particles. Ahmadzadeh and Ghanavati [14] have presented the PSO algorithm based navigation method for multiple mobile robots. The robots move according to the global best (g-best) position of a particle in every iteration. To prepare an optimal intelligent controller for an autonomous wheeled mobile robot, the Castillo et al. [120] have designed the hybridization of an Ant Colony Optimization (ACO) algorithm and the Particle Swarm Optimization (PSO) algorithm to optimize the membership function of a fuzzy controller. Zhang et al. [121] have proposed the Multi-Objective Particle Swarm Optimization Algorithm (MOPSO) to search a collision-free optimal path in the uncertain dynamic environment. Zhang and Li [122] have presented a new objective function for mobile robot navigation using PSO. This objective function works based on the position of the obstacles and target in the environment. PSO algorithm has been successfully applied by Raja and Pugazhenthi [123] to optimize the travel time of the mobile robot in the dynamic environments. This algorithm searches the feasible path in the environment by randomly in every iteration. Masehian and Sedighizadeh [124] have solved the motion planning problem of the mobile robot by using multi-objective PSO.

PSO-based optimal fuzzy controller has been designed by Wong et al. [125] to determine the velocities of the left-wheeled motor and right-wheeled motor of the differential drive mobile robot. Specialized particle swarm optimization algorithm has been presented by Li et al. [126] for global optimum path planning of mobile robots. The authors have conducted many simulation tests in the simple and complicated environment to show the effectiveness of the proposed algorithm. Huang [127] has designed the Parallel Metaheuristic Particle Swarm Optimization (PPSO) algorithm to solve the global path planning problem of an autonomous mobile robot. The author has implemented this PPSO algorithm in real-time using the field-programmable gate array (FPGA) chip. Chung et al. [128] have developed PSO and fuzzy based combinatorial algorithm to design intelligent navigation architecture for a mobile robot. They have used PSO algorithm to escape the robot from the dead-end condition, and the fuzzy algorithm is used to control the turn angle of a wheeled mobile robot during navigation and obstacle avoidance. Shiltagh and Jalal [129] have investigated the application of Modified Particle Swarm Optimization (MPSO) in the field of mobile robotics to determine a shortest feasible path from the beginning to end in an environment between obstacles. The developed modified PSO increases the convergence rate of the algorithms. Chatterjee and Matsuno [130] have solved the Simultaneous Localization and Mapping (SLAM) problem of mobile robots or vehicle using modified PSO and fuzzy evolutionary algorithm. Juang and Chang [131] have presented an evolutionary-group-based particle-swarm-optimization (EGPSO) for automatic learning of fuzzy system for mobile robot navigation or wall-following control in unknown environments. In [132], the authors have converted the robot path planning problem to the minimization problem and designed a fitness function based on the positions of the target and obstacles in the environment. Allawi and Abdalla [133] have proposed the sensor based PSO-fuzzy type-2 model for the navigation of multiple mobile robots. They have used PSO algorithm to determine the optimal input/output membership function parameters and rules for the fuzzy type-2 controller.

2.3.7 Ant Colony Optimization Algorithm and Other Nondeterministic Algorithms for Mobile Robot Navigation

The Ant Colony Optimization (ACO) algorithm is used by many authors for mobile robot navigation and obstacle avoidance in the different environments. ACO is a probabilistic algorithm proposed by Dorigo et al. [134] in 1999, which is originated from bionics. Guan-Zheng et al. [135] have presented the modern global path planning method for a mobile robot by applying Ant Colony System (ACS) algorithm and the Dijkstra algorithm. Purian and Sadeghian [136] have explored the optimal path for a mobile robot in an unknown dynamic environment using Ant Colony Optimization (ACO) algorithm and fuzzy controller. This ACO algorithm searches the optimal value from the fuzzy rule table and minimizes the distance between the start point to goal point of the mobile robot with obstacle avoidance competence. Bi et al. [137] have designed an Ant Colony System (ACS) to improve the path searching speed of the mobile robot in the dynamic environment. Dong et al. [138] have presented an improved ACO algorithm for obstacle avoidance of mobile robot in the grid environment. In [139], the authors have described various behaviors such as goal-seeking, wall-following and obstacle avoidance for mobile robot navigation using improved ACO algorithm. Fan et al. [140] have applied an intensified ant colony optimization (ACO) algorithm to search an optimal path for mobile robot between irregular obstacles in an environment. Sariff and Buniyamin [141] have compared the performances of GA and ACO algorithm for robot path planning in the global static environment and stated that the ACO algorithm takes less time to search an optimal path in the environment compared to GA. Hsu et al. [142] have proposed an improved ant colony system algorithm by including a new pheromone updating parameter for path planning of mobile robots. Ganganath et al. [143] have designed an off-line path planner for nonholonomic mobile robots using an ACO algorithm. Juang and Hsu [144] have designed the reinforcement ant optimized fuzzy controller (RAOFC) and applied it for wheeled mobile robot wall-following control under reinforcement learning environments. The inputs of the proposed controller are range-finding sonar sensors, and the output is a robot steering angle. The antecedent and consequent parts of the fuzzy controller have aligned by the fuzzy type-2 clustering and ACO respectively.

Hsu and Juang [145] have designed the wall-following mobile robot using a type-2 fuzzy controller (IT2FC) and integrated it with an ACO algorithm to improve the performance of the controller. The steering angle and moving speed of the wall-following mobile robot has been controlled by two type-2 fuzzy controllers. In [146], the authors have presented the navigation method of the two robots (a leader robot and a follower robot) using fuzzy controllers (FC). They have applied continuous ant colony optimization and particle swarm optimization (AF-CACPSO) to the control the mobile robots to perform obstacle boundary following behavior. Hsu and Juang [147] have adopted the multi-objective ACO for optimized the rule parameters of the fuzzy controller (FC) for wall-following mobile robot. Chen et al. [148] have designed a scent pervasion (pheromone) principle of ant (ACO) based robotic path planning in a map environment. Hossain and Ferdousand [149] have applied Bacterial Foraging Optimization (BFO) method for mobile robot navigation to find out shortest possible path within the minimum time from the start position to the goal position between moving obstacles. Liang et al. [150] have developed a bacterial foraging algorithm for making a bio-inspired path planning strategy for a mobile robot. In the proposed model, the behavior of bacteria is applied to search an optimal collision-free path between the start nodes to the target node in an environment with obstacles. Brand and Yu [151] have applied the Firefly Algorithm (Glowworm swarm optimization) to find a collision free shortest path in the two-dimensional static and dynamic environment for a mobile robot. They have compared this proposed algorithm to ACO algorithm and stated that the proposed algorithm provides better results (in terms of path length and computational cost) compared to ACO algorithm. Mohajer et al. [152] have presented a new Random Particle Optimization Algorithm (RPOA), which is inspired by the bacterial foraging technique, and used for local path planning for mobile robots in the dynamic and unknown environments. The proposed algorithm randomly searches the feasible path in the environment and avoids the moving obstacles by using the sensors.

2.3.8 Wind Driven Optimization Algorithm

Wind Driven Optimization (WDO) is a new type population-based iterative Nondeterministic optimization algorithm, which is inspired from the earth's atmosphere and developed by Bayraktar et al. [153, 154]. The authors have introduced and applied this WDO algorithm to solve the various electromagnetics optimization problems such as synthesis of a linear antenna array, double-sided artificial magnetic conductor, and E-shaped microstrip patch antenna. Furthermore, the authors have compared the results of proposed algorithm to other developed algorithm such as PSO, GA, and Differential Evolution (DE) and stated that the proposed algorithm performs better as compared to other developed algorithm. In [155], the authors have designed an optimal high-impedance Metasurfaces with Ultrasmall interwoven unit cells using WDO algorithm. Kuldeep et al. [156] have applied various nature inspired optimization algorithms such as Cuckoo Search (CS), Modified Cuckoo Search (MCS), WDO, PSO, and Artificial Bee Colony (ABC) algorithms to optimize the 2-channel linear phase quadrature mirror filter

(QMF) bank design. Bhandari et al. [157] have employed CS and WDO to achieve optimal or near-optimal threshold values of the satellite image segmentation. In an article [158], the authors have improved the performance of WDO algorithm using Levy flights. They have compared their performance using developed benchmark functions and stated that in some cases Levy flights based WDO gives better results compared to standalone WDO algorithm. Boulesnane and Meshoul [159] have presented modified WDO called as Multi-Region Modified WDO (MR-MWDO) to improve the impact of pressure on velocities of particles. The authors have solved the dynamic optimization problems of the moving peaks benchmark function using developed MR-MWDO algorithm.

2.4 Summary

This chapter provides a literature survey of various techniques employed for mobile robot navigation. After summarizing the above literature review, the major conclusions are listed below: -

1) The various soft computing techniques e.g. Deterministic, Nondeterministic, and Evolutionary algorithms, etc. have been applied by the researchers for mobile robot navigation and obstacle avoidance in the different environments.
2) According to literature survey, most of the researchers have used these soft computing techniques for mobile robot navigation and obstacle avoidance in only static environments. However, few researchers have considered dynamic environments for mobile robot navigation.
3) From the literature survey, it is observed that many researchers have demonstrated only computer simulation results without implementations of physical robot.
4) Nature-inspired algorithm based mobile robot navigation and obstacle avoidance is an important topic for the research. The hybridization of Deterministic and Nondeterministic algorithms is also a better choice for the research.

Motivated by the aforementioned literatures, the present research work focuses on the design and implementation of robust techniques, which can efficiently solve the navigation and obstacle avoidance problems of the mobile robot in static and dynamic environments.

Chapter 3

Kinematic and Dynamic Analysis of the Nonholonomic Differential Drive Wheeled Mobile Robot

3.1 Introduction

Nowadays, the wheeled mobile robots are widely used to carry out many tasks such as planetary exploration (e.g. Mars rover), military applications (e.g. bomb disposal mobile robot), and industrial applications (e.g. mobile manipulators), etc. Therefore, it is important to know the kinematic and dynamic characteristics of the wheeled mobile robots. This chapter describes the kinematic analysis of the nonholonomic differential drive two-wheeled mobile robot. The analysis of the nonholonomic differential drive two-wheeled mobile robot consists of two models, namely kinematic analysis and dynamic analysis [160]. In the kinematic analysis, the motion of the robot is studied without considering the affecting forces. On the other hand, in the dynamic analysis, we have studied the various forces, which are responsible for this motion of the robot. The rest of this chapter is organized as follows: Section 3.2 introduces the different kinematic equations of the nonholonomic differential drive two-wheeled mobile robot. Section 3.3 presents the description of dynamic equations of the mobile robot. Finally, Section 3.4 depicts the summary.

3.2 Kinematic Model of the Nonholonomic Differential Drive Two-Wheeled Mobile Robot

Figure 3.1 illustrates the kinematic and dynamic model of the nonholonomic differential drive two-wheeled mobile robot. The term nonholonomic means the robot cannot move sideways, and it moves based on the principle of rolling wheels [161]. The robot consists of the two front driving wheels and one caster wheel for carrying the chassis. The two separate motors have been used for driving the wheels and control the motion and orientation of the robot. It is assumed that the robot has been made by a rigid frame, and it is moving in a horizontal plane. In Figure 3.1, L, R, and C denotes the track width, radius of the wheels, and center of the mass of a mobile robot, respectively. The point P is located between the centres of the driving wheels axis. The point d is the distance between the points P and C. The landmark (O, X, Y) shows the field navigation environment, and (O, x, y) is the moving axis of the mobile robot. The θ is the turning angle, which represents the orientation of the robot about an axis (O, X). The three parameters (x, y, θ) describe the initial posture of the mobile robot, which is denoted by q :-

$$q = [x, y, \theta]^T \tag{3.1}$$

The mobile robot follows a nonholonomic constraint that means the driving wheels purely roll without slipping. The nonholonomic constraint is written as the following equation: -

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \tag{3.2}$$

The following equation describes the relationship between the linear velocity and angular velocity of the wheels: -

$$V = \omega \cdot R \tag{3.3}$$

$$V_R = \omega_R \cdot R \tag{3.4}$$

$$V_L = \omega_L \cdot R \tag{3.5}$$

$$\omega = \frac{V_R - V_L}{L} \tag{3.6}$$

$$V = \frac{V_R + V_L}{2} \tag{3.7}$$

The equations (3.6) and (3.7) can be written as follows: -

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} V_R \\ V_L \end{bmatrix} \tag{3.8}$$

The following equation calculates the moving axis (x, y) velocities of the mobile robot and its turning (orientation) angle with respect to time (t): -

$$\frac{dx}{dt} = \dot{x} = V \cdot \cos\theta \tag{3.9}$$

$$\frac{dy}{dt} = \dot{y} = V \cdot \sin\theta \tag{3.10}$$

$$\frac{d\theta}{dt} = \dot{\theta} = \omega \tag{3.11}$$

The kinematic equations of the two-wheeled mobile robot are as follows: -

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix} \tag{3.12}$$

The equations (3.9), (3.10) and (3.11) are updated by equations (3.6) and (3.7): -

$$\dot{x} = \frac{R}{2}(\omega_R + \omega_L) \cdot \cos\theta \tag{3.13}$$

$$\dot{y} = \frac{R}{2}(\omega_R + \omega_L) \cdot \sin\theta \tag{3.14}$$

$$\dot{\theta} = \frac{R}{L}(\omega_R - \omega_L) \tag{3.15}$$

Kinematic model of the mobile robot is obtained by the combining of the equations (3.13), (3.14), and (3.15): -

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2}\cos\theta & \frac{R}{2}\cos\theta \\ \frac{R}{2}\sin\theta & \frac{R}{2}\sin\theta \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \tag{3.16}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{\cos\theta}{2} & \frac{\cos\theta}{2} \\ \frac{\sin\theta}{2} & \frac{\sin\theta}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} V_R \\ V_L \end{bmatrix} \tag{3.17}$$

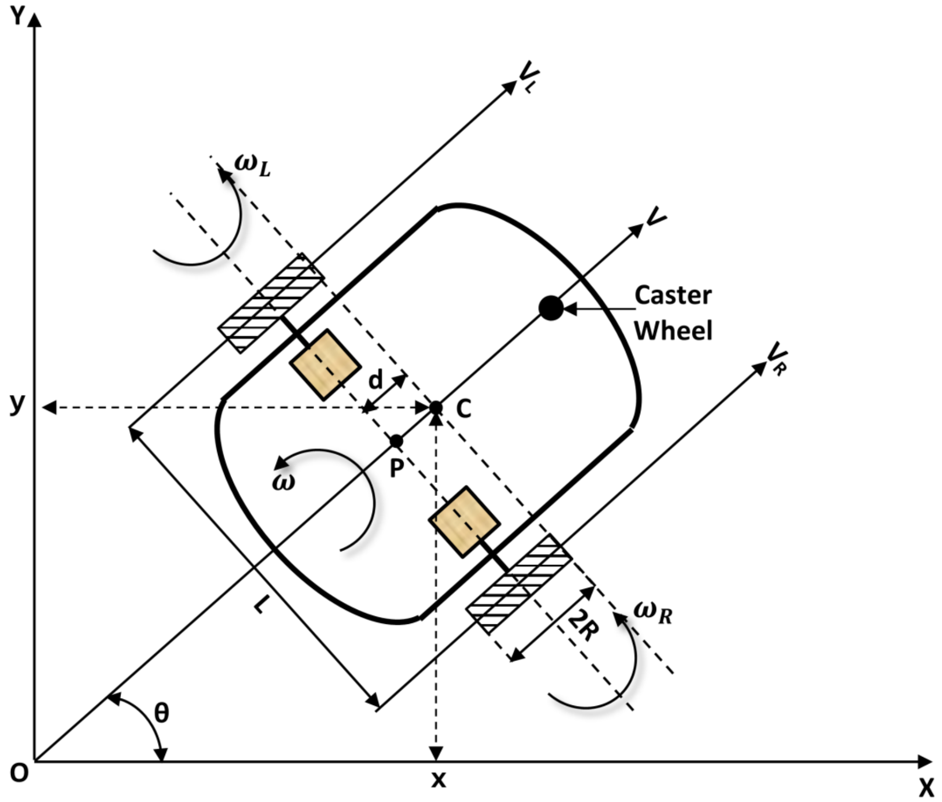


Figure 3.1: Kinematic and dynamic model of the nonholonomic differential drive two-wheeled mobile robot.

where V_R and V_L are the linear velocities of the right and left wheels, respectively, which are used as a motion command to the motors (wheels) for mobile robot navigation and obstacle avoidance. Similarly, ω_R and ω_L are the angular velocities of the right and left wheels, respectively. The V and ω denote the centre linear (forward) velocity and centre angular (rotational) velocity of the mobile robot, respectively.

The following conditions are used to control the motion and orientation of the mobile robot: -

(1) If $V_L = V_R$, then the robot moves straight (Figure 3.2).

(2) If $V_L < V_R$, then the robot turns left side (Figure 3.3).

(3) If $V_L > V_R$, then the robot turns right side (Figure 3.4).

(4) If $V_L = -V_R$, then the robot rotates (spin) clockwise (Figure 3.5).

(5) If $-V_L = V_R$, then the robot rotates (spin) anticlockwise (Figure 3.6).

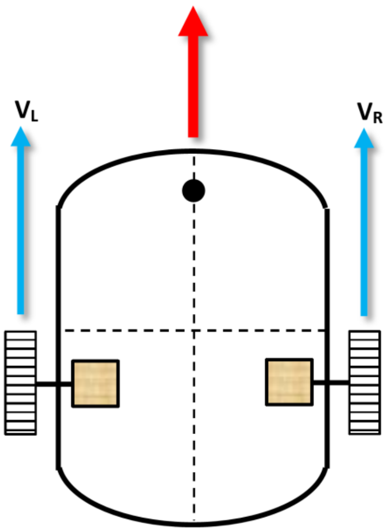


Figure 3.2: Robot moves straight $(V_L = V_R)$.

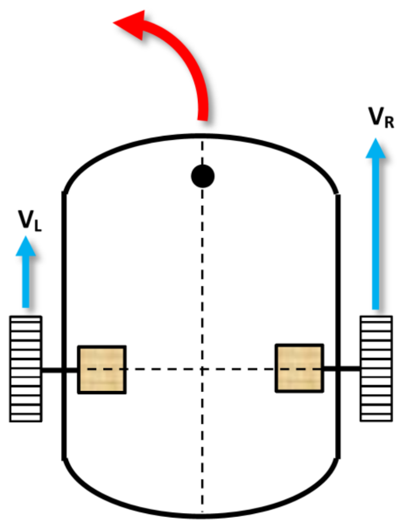


Figure 3.3: Robot turns left side $(V_L < V_R)$.

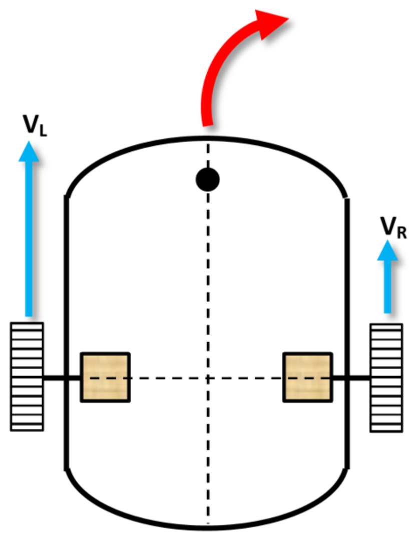


Figure 3.4: Robot turns right side $(V_L > V_R)$.

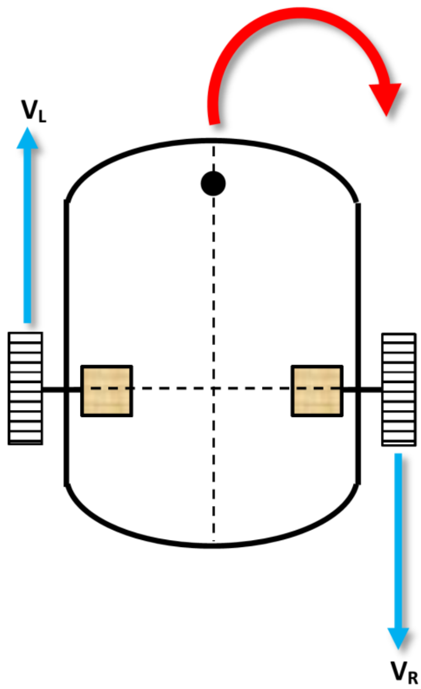


Figure 3.5: Robot rotates clockwise $(V_L = -V_R)$.

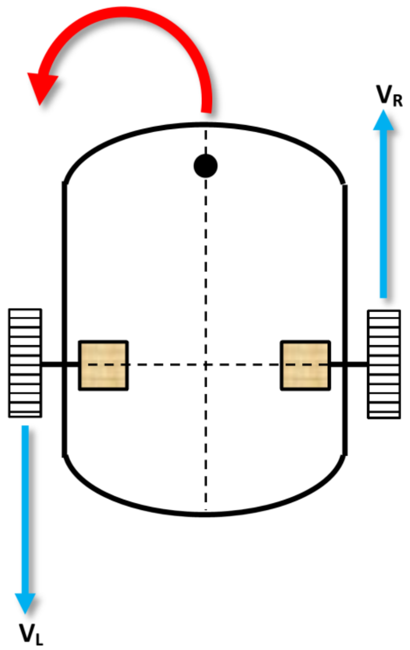


Figure 3.6: Robot rotates anticlockwise $(-V_L = V_R)$.

3.3 Dynamic Model of the Nonholonomic Differential Drive Two-Wheeled Mobile Robot

This section presents the dynamic model of the nonholonomic differential drive two-wheeled mobile robot. Forces must be applied to the mobile robot to generate motion [23]. These motions are used for studying the dynamic model of the mobile robot. According to the Euler-Lagrangian formulation, the dynamic model of the nonholonomic differential drive two-wheeled mobile robot can be described as follows [21, 23, 162]:-

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+\tau_d = B(q)\tau - A^T(q)\lambda \tag{3.18}$$

where $M(q)\in\Re^{n\times n}$ is a symmetric positive definite inertia matrix, $C(q,\dot{q})\in\Re^{n\times n}$ is the centripetal and Coriolis matrix, $\tau_d \in\Re^{n\times 1}$ indicates the norm-bounded unknown external disturbance vector, $B(q)\in\Re^{n\times r}$ is the input transformation matrix, $\tau\in\Re^{r\times 1}$ is the torque vector, $A^T(q)\in\Re^{n\times m}$ is the matrix associated with the constraints, and $\lambda\in\Re^{m\times 1}$ is the vector constraint forces.

$$M(q)=\begin{bmatrix} m & 0 & md\sin\theta \\ 0 & m & -md\cos\theta \\ md\sin\theta & -md\cos\theta & I \end{bmatrix} \tag{3.19}$$

$$\ddot{q}=\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} \tag{3.20}$$

$$C(q,\dot{q})=\begin{bmatrix} 0 & 0 & md\dot{\theta}\cos\theta \\ 0 & 0 & md\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix} \tag{3.21}$$

$$B(q)=\frac{1}{R}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ \frac{L}{2} & -\frac{L}{2} \end{bmatrix} \tag{3.22}$$

$$\tau = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \qquad (3.23)$$

$$A^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ -d \end{bmatrix} \qquad (3.24)$$

$$\lambda = -m(\dot{x}\cos\theta + \dot{y}\sin\theta)\dot{\theta} \qquad (3.25)$$

where m is the total mass of the mobile robot. I is the moment of inertia of the mobile robot. τ_R and τ_L represent the right and left wheel (motor) torques, respectively.

3.4 Summary

This chapter introduces the kinematic and dynamic equations of the nonholonomic differential drive two-wheeled mobile robot. The major contributions of this chapter are summarized as follows: -

- The kinematic and dynamic analysis of the mobile robot provides the knowledge about its positions in the landmark (O, X, Y).
- Before applying soft computing techniques for mobile robot navigation and obstacle avoidance, it is important to know the kinematic and dynamic parameters of the wheeled mobile robot.
- Using these right and left wheel velocity equations, the motion and orientation of the mobile robot can be controlled during navigation in the environment.
- Five conditions have been described to control the steering of the nonholonomic differential drive mobile robot for obstacle avoidance.

Chapter 4

Intelligent Navigation of a Mobile Robot in Static and Dynamic Environments using Hybrid Fuzzy Architecture

4.1 Introduction

Navigation and obstacle avoidance are one of the most important tasks for any mobile robot. This chapter proposes the design and implementation of a hybrid fuzzy (H-Fuzzy) architecture for intelligent navigation of a mobile robot in the static and dynamic environments. The proposed H-Fuzzy architecture is the combination of Takagi-Sugeno type and Mamdani-type fuzzy logics, which helps the mobile robot to reach the goal with obstacle avoidance. The Takagi-Sugeno type fuzzy logic architecture (TFa) is used to assist the robot to reach the goal. The inputs of the Takagi-Sugeno type fuzzy logic architecture are obstacle distances received from the group of sensors, and the output is the turning angle between the robot and the goal. The Mamdani-type fuzzy logic architecture (MFa) is integrated with the TFa to control the motor velocities of the robot. Computer simulations are conducted through MATLAB software and implemented in real time by using Arduino microcontroller based wheeled mobile robot. Moreover, the successful experimental results on the actual mobile robot demonstrate the superiority of the proposed architecture.

Over the last two-three decades, various techniques have been employed for mobile robot navigation and obstacle avoidance. However, the most of the researchers have applied fuzzy logic techniques for solving these types of the problem because there is a unique feature in fuzzy technique to handle the system uncertainty without large computation model. As compared to other soft computing techniques, the fuzzy logic system is easy to understand. The concept of fuzzy logic has been introduced by Zadeh

[25], which is extensively used in many engineering applications such as mobile robotics. The structure of fuzzy logic has been created by human knowledge, or it can be generated from the dataset. The other soft computing techniques such as Neural Network (NN), Genetic Algorithm (GA), Particle Swarm Optimization (PSO) algorithm, Simulated Annealing Algorithm (SAA), and Ant Colony Optimization (ACO) algorithm have been widely integrated with this fuzzy logic to improve its performance. For autonomous mobile robot navigation, it is important to control the orientation and speed of the mobile robot using sensor data. In [144, 146], the authors have been designed and implemented an evolutionary fuzzy controller for mobile robot wall-following control.

This chapter utilizes combined fuzzy architecture, called as H-Fuzzy architecture for intelligent mobile robot navigation. The proposed H-Fuzzy architecture aims to control the turning angle (between the robot and goal) and motor velocities of a mobile robot from start point to goal point with obstacle avoidance competence. This architecture is tested in various simulation and experimental environment and is found to be a good agreement. Rest of the chapter is organized as follows: Section 4.2 introduces the design and implementation of the H-Fuzzy architecture for intelligent navigation of the mobile robot in the static and dynamic environments. Section 4.3 demonstrates the computer simulation results in different unknown environments. Section 4.4 describes the simulation result comparison with previous works. Section 4.5 presents the experimental results and discussion for validating the proposed controller. Finally, Section 4.6 depicts the summary.

4.2 Hybrid Fuzzy (H-Fuzzy) Architecture

This section introduces the design and implementation of the H-Fuzzy architecture for intelligent navigation of a mobile robot in the static and dynamic environments. Takagi-Sugeno type fuzzy logic architecture (TFa) is used to assist the robot to reach the goal, and the Mamdani-type fuzzy logic architecture (MFa) is used to control the right motor velocity and left motor velocity of the mobile robot. Both the fuzzy logic architectures (TFa and MFa) receive input (obstacle distances) from a group of sensors to control the turning angle and motor velocities of the mobile robot during navigation. Figure 4.1

illustrates the proposed architecture of the hybrid fuzzy logic for intelligent mobile robot navigation.

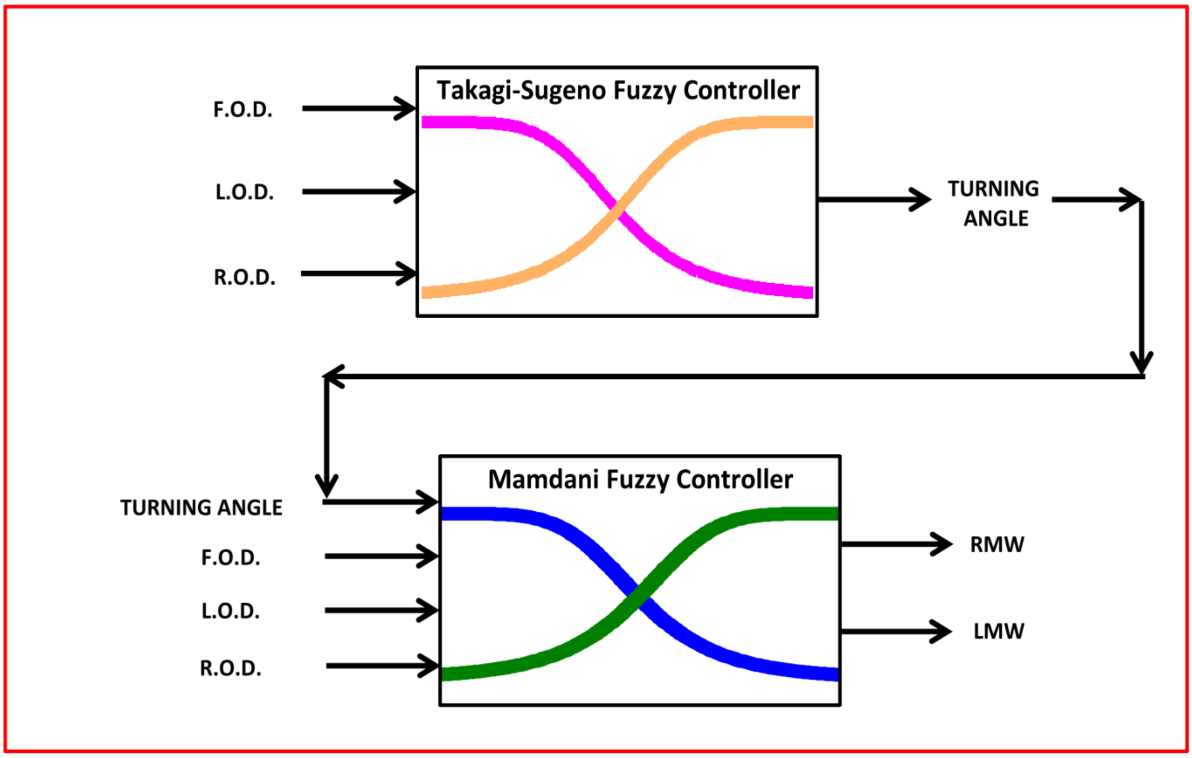


Figure 4.1: The proposed architecture of hybrid fuzzy (H-Fuzzy) logic for intelligent mobile robot navigation.

4.2.1 Takagi-Sugeno Type Fuzzy Logic Architecture (TFa) for Goal Reaching

In this section, the Takagi-Sugeno type fuzzy logic architecture (TFa) is used to assist the mobile robot to reach the goal in the static and dynamic environments. The proposed TFa has three inputs and single output. The TFa receives these inputs (obstacle distances) from the front, left, and the right group of sensors of the robot. These sensors read the obstacle from 20cm to 150cm approximately. So, the ranges of inputs are divided between 20cm to 150cm. These inputs are denoted by F.O.D. (Front Obstacle Distance), L.O.D. (Left Obstacle Distance), and R.O.D. (Right Obstacle Distance), respectively. The two generalized bell-shaped (Gbell) linguistic variables, namely CLOSE and AWAY, respectively, are selected for F.O.D., L.O.D., and R.O.D. The output of this TFa is a turning angle (T.A.) between the robot and goal. The two constant type linguistic variables NEGATIVE and POSITIVE respectively have been selected for the output, and it is located at -90 and 90 respectively. Table 4.1 describes the fuzzy rule set of the TFa, which helps the robot to reach the goal in the different environments. Figure 4.3 illustrates the general structure of the Takagi-Sugeno Type Fuzzy Logic Architecture (TFa). Figures 4.4 and 4.5 show the linguistic variables of the inputs and output, respectively. This fuzzy architecture is composed through zero-order Takagi-Sugeno model in the following form: -

$$Rule_n : \text{ If } p_1 \text{ is } X_{i1},\ p_2 \text{ is } X_{i2},\ \text{And } p_3 \text{ is } X_{i3} \text{ THEN } f_i \text{ is } \alpha_i \tag{4.1}$$

where n=1, 2, 3…8 (eight rules), the symbols p_1, p_2, and p_3 are the input variables. The X_{i1}, X_{i2}, and X_{i3} are fuzzy sets of the input variables, and α_i is a real number. The i=1, 2, because each input has two Gbell membership functions. The fuzzy set X_{ij} uses the following generalized bell-shaped linguistic variables: -

$$\mu_{ij}\left(p_j\right) = \frac{1}{1+\left|\dfrac{p_j - c_{ij}}{a_{ij}}\right|^{2b_{ij}}} \tag{4.2}$$

where j=1…3 (three inputs), the c_{ij}, a_{ij}, and b_{ij} are the center, half width, and slope controlling parameters of each generalized bell-shaped membership function, respectively. The basic structure of the generalized bell-shaped membership function is shown in Figure 4.2. The firing strength $\delta_i(p)$ has calculated by the following function: -

$$\delta_i(p) = \prod_{j=1}^{n} \mu_{ij}(p_j) \tag{4.3}$$

The defuzzification of the output variable (turning angle) is calculated by the weighted average method: -

$$f_i = \frac{\sum_{n=1}^{8} \delta_i(p)\cdot\alpha_i}{\sum_{n=1}^{8} \delta_i(p)} \tag{4.4}$$

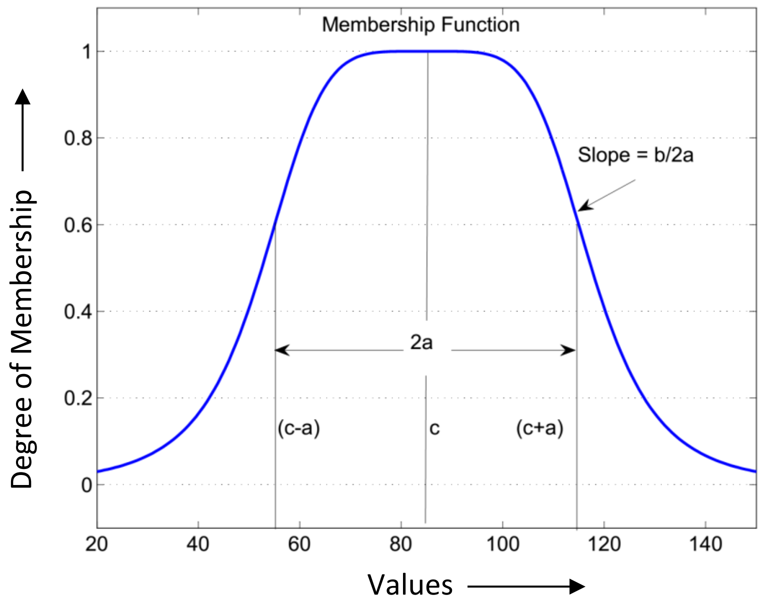


Figure 4.2: The basic structure of the generalized bell-shaped membership function.

Table 4.1: Fuzzy rule set of the Takagi-Sugeno type fuzzy logic architecture (TFa)

Fuzzy rules	**F.O.D. (cm)**	**L.O.D. (cm)**	**R.O.D. (cm)**	**T. A. (degree)**
1	Away	Away	Away	Positive
2	Close	Close	Close	Negative
3	Away	Close	Away	Negative
4	Away	Away	Close	Positive
5	Close	Away	Away	Negative
6	Close	Close	Away	Negative
7	Close	Away	Close	Positive
8	Away	Close	Close	Positive

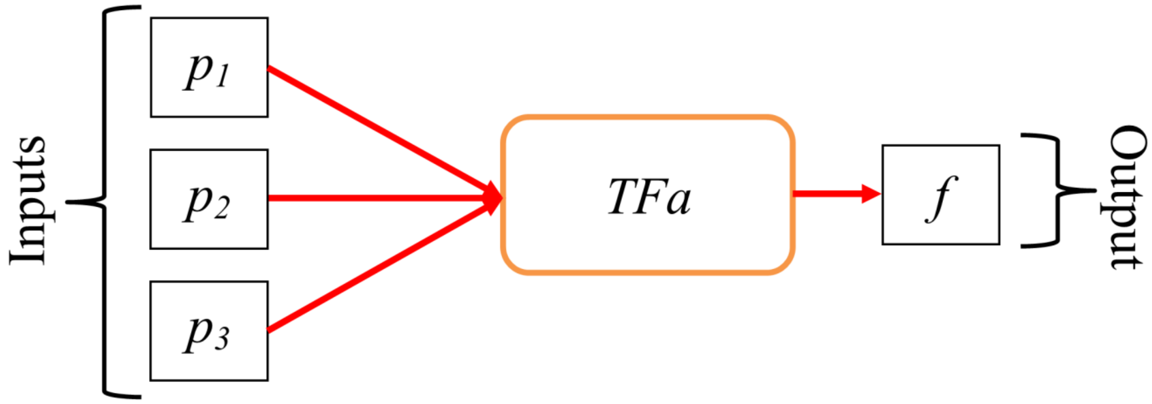


Figure 4.3: The general structure of the Takagi-Sugeno type fuzzy architecture (TFa).

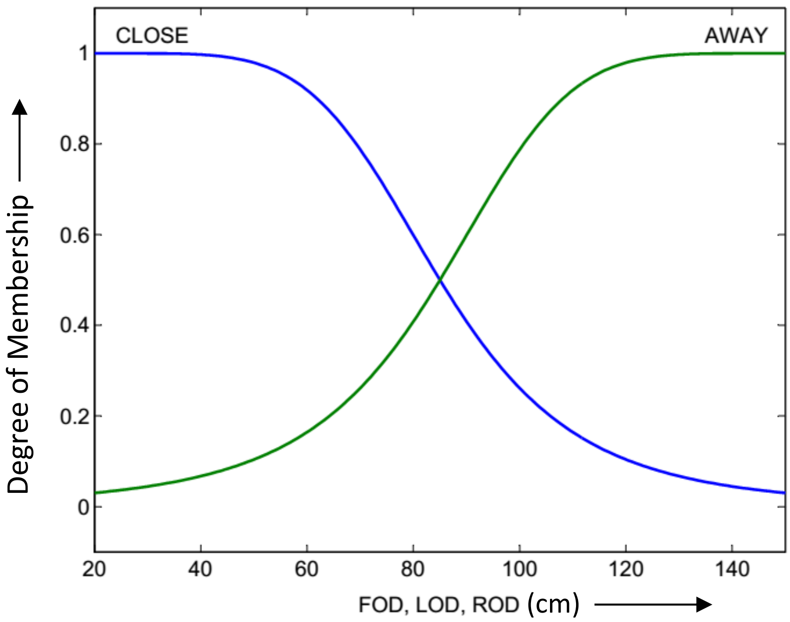


Figure 4.4: The membership functions of the input variables (F.O.D., L.O.D., and R.O.D.).

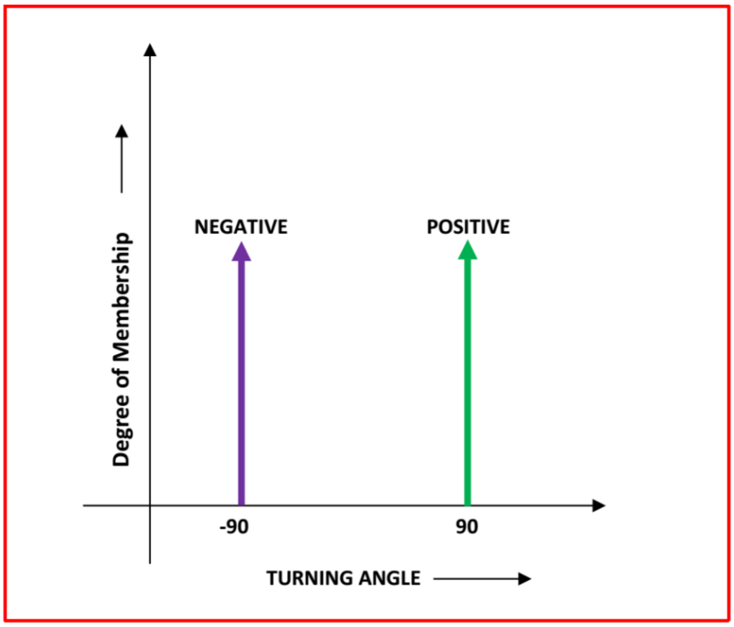


Figure 4.5: The constant type membership function of the output variable (Turning Angle).

4.2.2 Mamdani-Type Fuzzy Logic Architecture (MFa) for Obstacle Avoidance

This section introduces the design of Mamdani-type fuzzy logic architecture (MFa) for mobile robot navigation and obstacle avoidance in the static and dynamic environments. The proposed MFa is used to control the right motor velocity and left motor velocity of the mobile robot. The MFa has four inputs and two outputs. The MFa receives first three inputs (obstacle distance) from the group of sensors of the mobile robot. The first three inputs are denoted by F.O.D., L.O.D., and R.O.D., respectively. The fourth input is the turning angle (goal angle) between the robot and goal, which is received from the TFa. The outputs of the MFa are the velocities of the motors of a robot. The outputs are addressed by RMV (Right Motor Velocity) and LMV (Left Motor Velocity), respectively. The range of first three inputs is divided into two linguistic variables, namely CLOSE and AWAY, respectively, and it is located at 20cm to 150cm. The two linguistic variables NEGATIVE and POSITIVE, respectively, are used for turning angle (T.A.). The range of outputs is divided into two linguistic variables, namely LOW and HIGH, respectively. These outputs are located at 6.7cm/sec to 16.7cm/sec. The two generalized bell-shaped (Gbell) membership functions are selected for inputs and outputs. Figure 4.6 shows the input and output membership functions of the MFa. Figure 4.7 illustrates the fuzzy logic architecture. Table 4.2 describes the fuzzy rule set of the MFa. The MFa is composed through Mamdani-type fuzzy model in the following form: -

$$Rule_m : IF\ x_1\ is\ A_{j1},\ x_2\ is\ A_{j2},\ x_3\ is\ A_{j3},\ \&\ x_4\ is\ A_{j4}\ THEN\ y_1\ is\ B_{j1}\ \&\ y_2\ is\ B_{j2} \tag{4.5}$$

where m =1, 2, 3…12 (twelve rules), the x_1, x_2, x_3, and x_4 are the input variables. Similarly, y_1 and y_2 are the output variables. The A_{j1}, A_{j2}, A_{j3}, and A_{j4} are the fuzzy sets of the input variables. Similarly, B_{j1} and B_{j2} are the fuzzy sets of the output variables. The j =1, 2, because each input and output have two Gbell membership functions. The fuzzy set (inputs and outputs) uses the following Gbell membership function: -

$$\mu_{jk}(x_k; a, b, c) = \frac{1}{1 + \left| \frac{x_k - c_{jk}}{a_{jk}} \right|^{2b_{jk}}} \tag{4.6}$$

$$\mu_{jl}(y_l; a, b, c) = \frac{1}{1 + \left| \frac{y_l - c_{jl}}{a_{jl}} \right|^{2b_{jl}}} \tag{4.7}$$

where k =1…4 (four inputs), and l =1, 2 (two outputs). The symbols a_{jk}, b_{jk}, and c_{jk} are adjusting parameters of the Gbell membership function (see the Figure 4.2); called as the half width, slope control, and center, respectively.

The defuzzification of the output variables (y_1 and y_2) are accomplished by the weighted average method: -

$$y_1 = \frac{\sum_{m=1}^{12} \left(\mu_{j1}(x_1) \cdot \mu_{j2}(x_2) \cdot \mu_{j3}(x_3) \cdot \mu_{j4}(x_4) \right) \cdot y_1}{\sum_{m=1}^{12} \left(\mu_{j1}(x_1) \cdot \mu_{j2}(x_2) \cdot \mu_{j3}(x_3) \cdot \mu_{j4}(x_4) \right)} \tag{4.8}$$

$$y_2 = \frac{\sum_{m=1}^{12}\left(\mu_{j1}(x_1)\cdot\mu_{j2}(x_2)\cdot\mu_{j3}(x_3)\cdot\mu_{j4}(x_4)\right)\cdot y_2}{\sum_{m=1}^{12}\left(\mu_{j1}(x_1)\cdot\mu_{j2}(x_2)\cdot\mu_{j3}(x_3)\cdot\mu_{j4}(x_4)\right)} \tag{4.9}$$

Table 4.2: Fuzzy rule set of the Mamdani-type fuzzy logic architecture (MFa)

Fuzzy rules	**F.O.D. (cm)**	**L.O.D. (cm)**	**R.O.D. (cm)**	**T. A. (degree)**	**RMV (cm/sec)**	**LMV (cm/sec)**
1	Away	Away	Away	Positive	High	Low
2	Away	Away	Away	Negative	Low	High
3	Close	Close	Close	Negative	Low	High
4	Close	Close	Close	Positive	High	Low
5	Away	Close	Away	Negative	Low	High
6	Away	Away	Close	Positive	High	Low
7	Close	Away	Away	Negative	Low	High
8	Close	Away	Away	Positive	High	Low
9	Close	Close	Away	Negative	Low	High
10	Close	Away	Close	Positive	High	Low
11	Away	Close	Close	Positive	High	Low
12	Away	Close	Close	Negative	Low	High

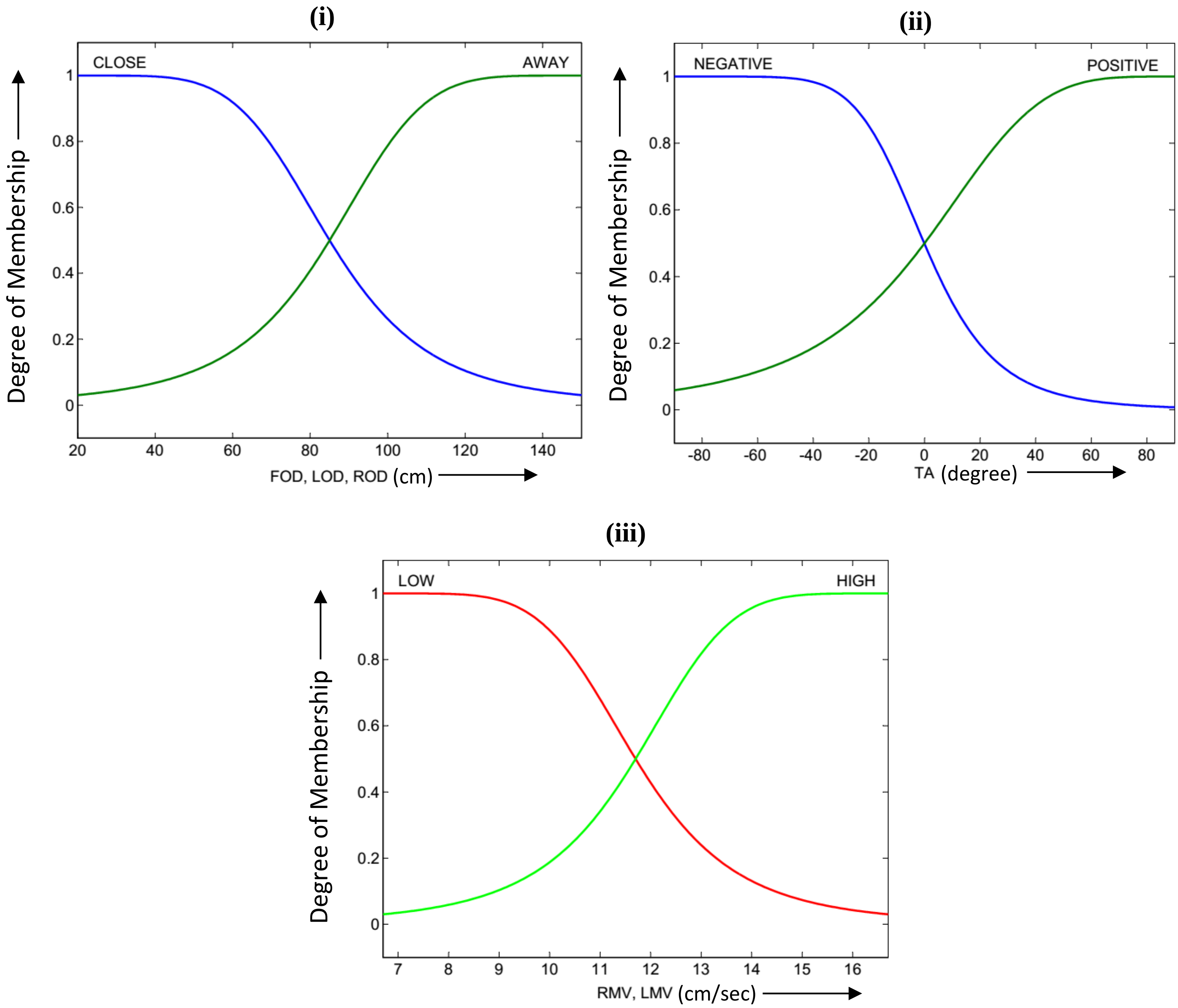


Figure 4.6: Membership functions **(i)** Obstacle distances (F.O.D., L.O.D. and R.O.D., respectively), **(ii)** Turning angle (T.A.), and **(iii)** Motor velocities (Right and Left, respectively)

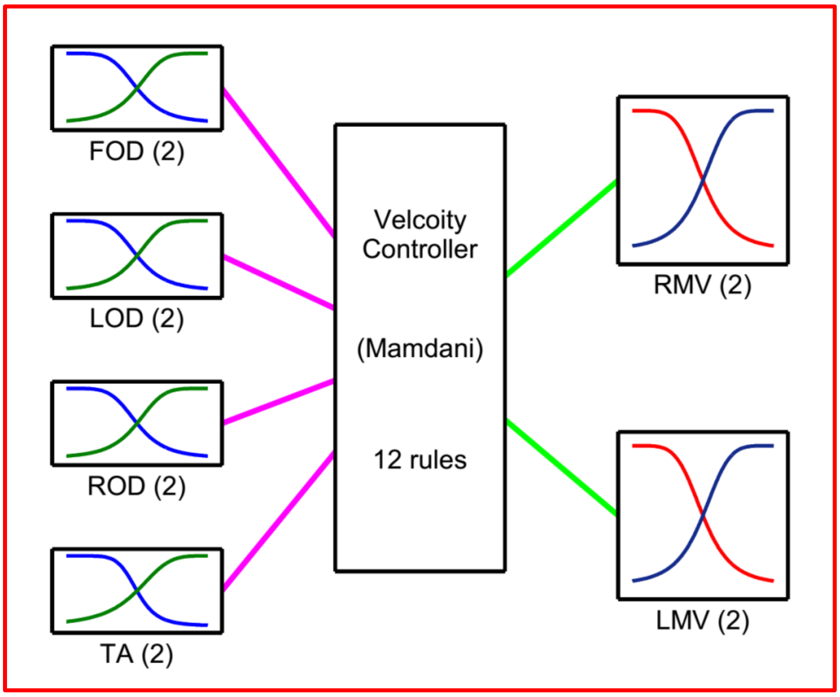


Figure 4.7: Fuzzy logic architecture.

4.3 Simulation Studies

This section illustrates the successful simulation results of the mobile robot navigation in a different type of environments. The simulations are conducted through the MATLAB software on the HP desktop and implemented in real time by using Arduino microcontroller based experimental mobile robot. The developed flowchart of mobile robot navigation based on H-Fuzzy architecture is given in Figure 4.8. Figures 4.9 and 4.10 show the simulation results of the mobile robot navigation in the unknown and indoor environment using H-Fuzzy architecture, respectively. In the simulation results, it is assumed that the position of the starting point and goal point is known, but the positions of all the obstacles in the environment are unknown for the robot. The width and height of the environment are 250cm and 250cm, respectively. When the obstacles are detected on the front, left, and right side of the robot, then the H-Fuzzy architecture is activated, and the robot adjusts its motor velocities (left and right) according to the architecture outputs.

Moreover, Figure 4.11 shows the mobile robot navigation in a typical U-shape complex environment. In the simulation test, the robot starts from the initial coordinate (10cm, 10cm) to the goal coordinate (230cm, 220cm) and there is a U-shaped complex obstacle located in the middle of the environment. It can be seen that from the simulation, when the robot comes near to the U-shaped obstacle, it turns right sharply and follows the wall to protect from the collision, and reach the goal successfully by avoiding the obstacle.

Figure 4.12 illustrates the simulation result of mobile robot navigation between the two moving obstacles. In the simulation, the first cyan color circular shape obstacle is moving from the coordinates (200cm, 150cm) to (59cm, 125cm) with the moving angle 190° from right to left. The second red color rectangular shape obstacle is moving from coordinates (150cm, 210cm) to (150cm, 103cm) with the moving angle 270° from up to down. From the simulation result, when the moving obstacles are near to the mobile robot, then the H-Fuzzy architecture is activated, and the robot turns right, i.e. the rate of Right Motor Velocity (RMV) decreases less than the rate of Left Motor Velocity (LMV), respectively. Table 4.3 illustrates the simulation results of mobile robot navigation in the different static and dynamic environments using H-Fuzzy architecture.

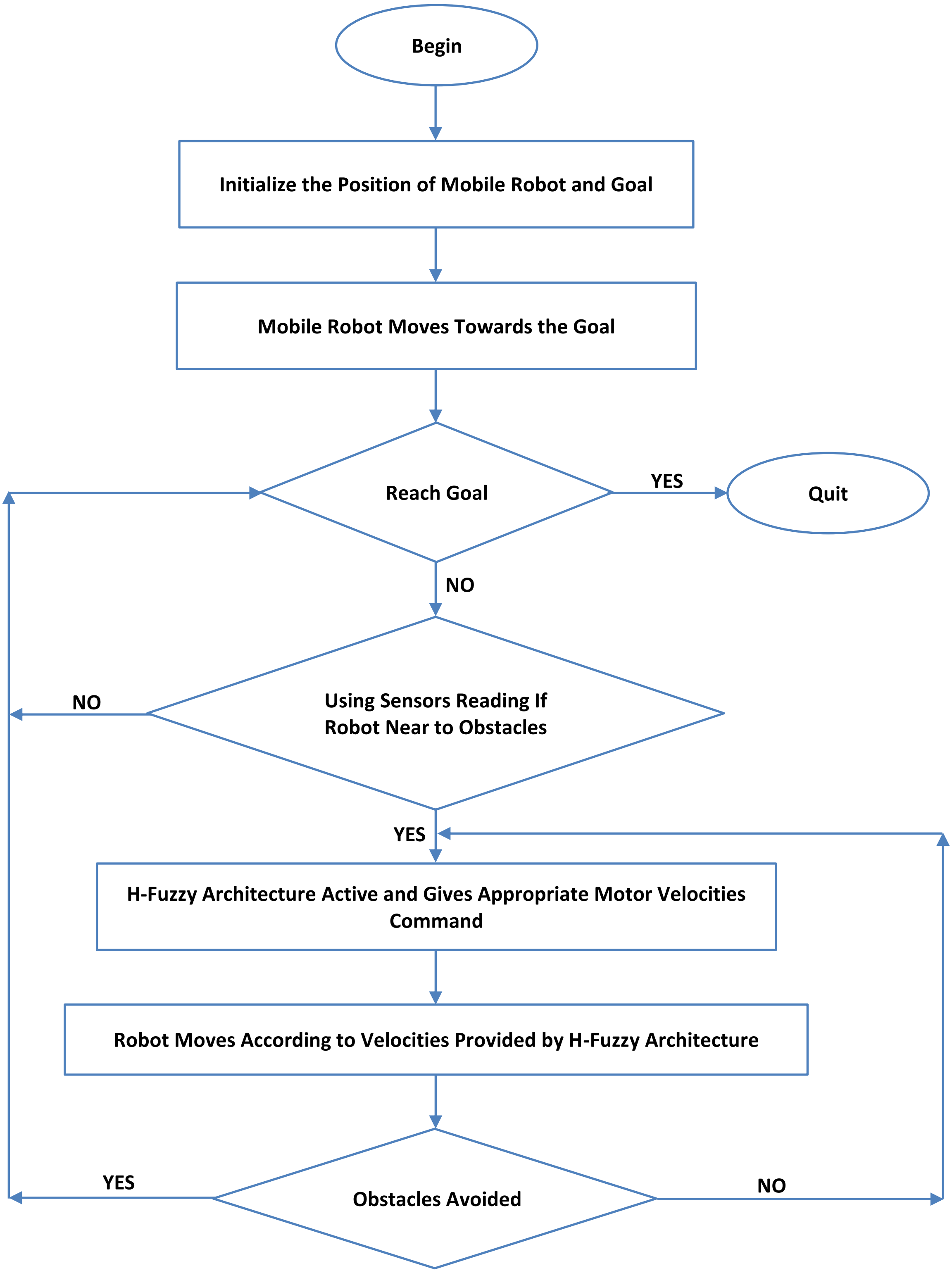


Figure 4.8: Flowchart of mobile robot navigation based on H-Fuzzy architecture.

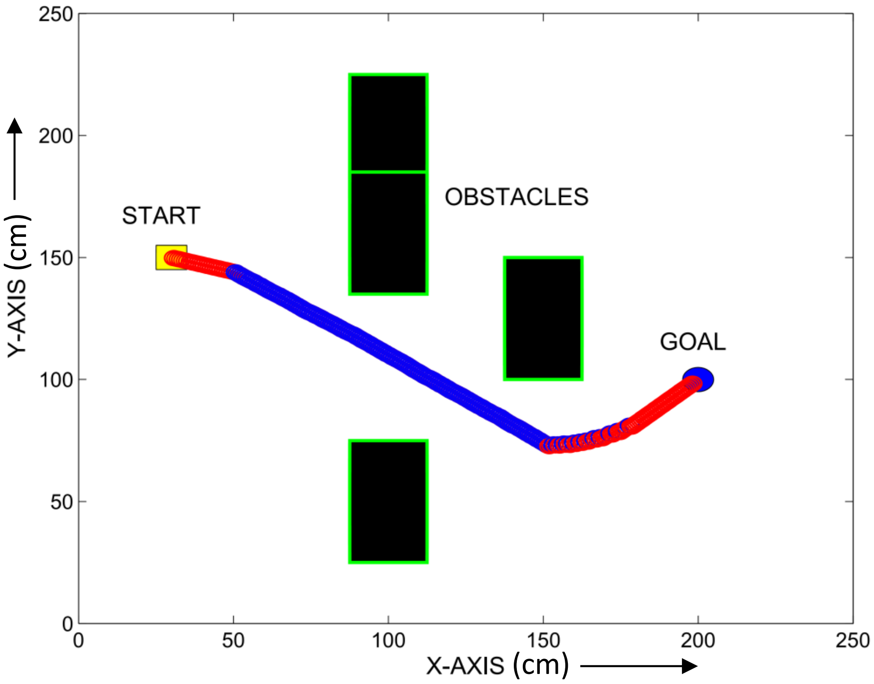


Figure 4.9: Navigation of a mobile robot in an unknown environment using H-Fuzzy architecture.

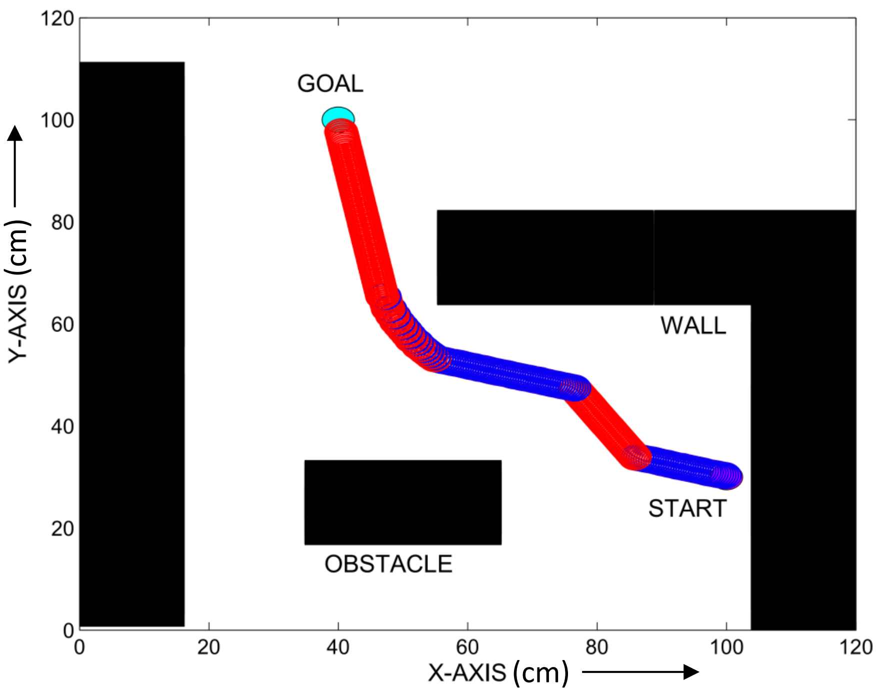


Figure 4.10: Navigation of a mobile robot in an indoor environment using H-Fuzzy architecture.

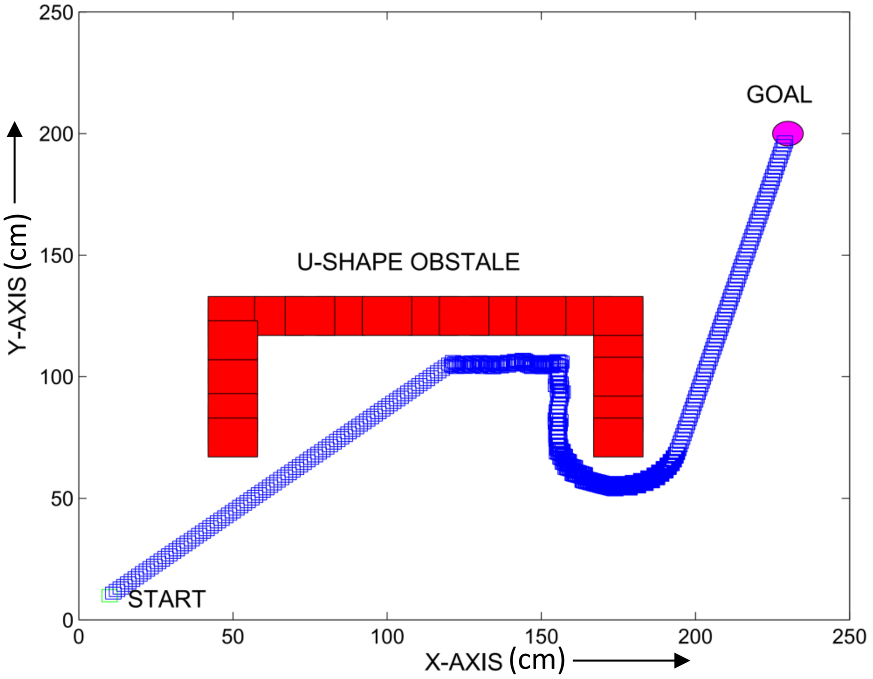


Figure 4.11: Navigation of a mobile robot in complex environment using H-Fuzzy architecture.

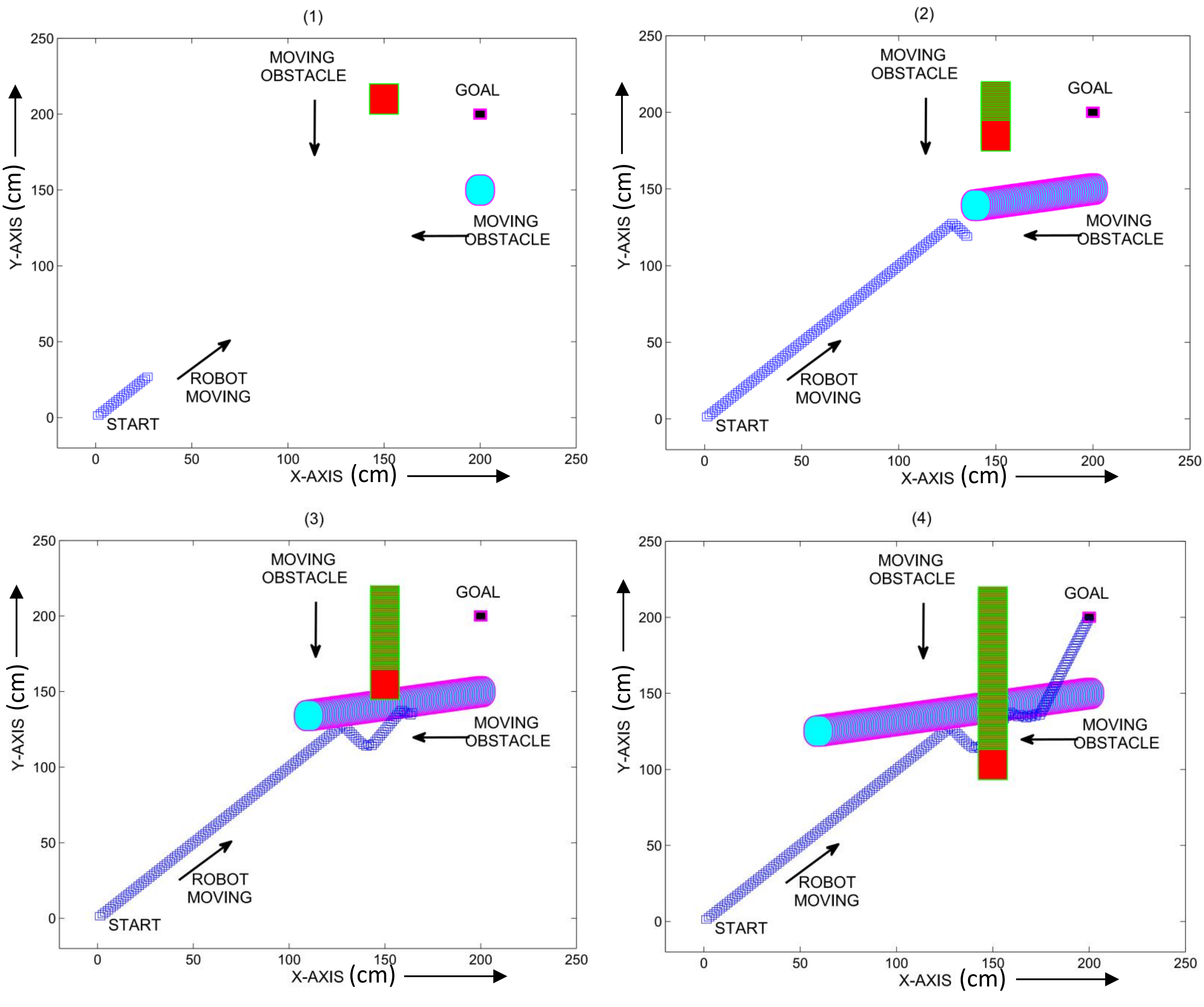


Figure 4.12: Navigation of a mobile robot in the dynamic environment using H-Fuzzy architecture.

Table 4.3: Simulation results of mobile robot navigation in the different static and dynamic environments using H-Fuzzy architecture

Figure no.	**Environment type**	**Travelling path length (cm)**	**Navigation time (sec)**
Figure 4.9	Unknown environment	73	9.2
Figure 4.10	Indoor environment	47	6.1
Figure 4.11	Complex environment	125	16.6
Figure 4.12	Dynamic environment	102	13.1

4.4 Comparison with Previous Works

This section describes the comparative study of proposed H-Fuzzy architecture over Fuzzy [163] and ANN (Artificial Neural Network) [53] methods in the simulation mode. In an article [163], the Cherroun and Boumehraz have prepared a Takagi-Sugeno type fuzzy behaviour based control system, which helps to navigate a tricycle type mobile robot autonomously in the unknown crowded environment. In [53], the authors have combined the multi-layer feed forward artificial neural network with Q-reinforcement learning method to construct a robust path-planning algorithm for the mobile robot.

The performance of this proposed architecture is evaluated by the path length and travelling path smoothness. Figure 4.13 and Figure 4.14 demonstrates the graphical comparison between the path made by Fuzzy [163] and ANN [53] over current developed H-Fuzzy architecture for the same path planning problems.

From the simulation analysis, it has been seen that the proposed architecture provides better trajectories in terms of path length and smoothness as compared to previous methods, and also it can efficiently drive the mobile robot in an optimal path in different environments. Tables 4.4 and 4.5 illustrate the path covered (in cm) by the robot to reach the goal using proposed architecture and previous methods [163, 53]. The centimeter measurements are taken on the proportional basis.

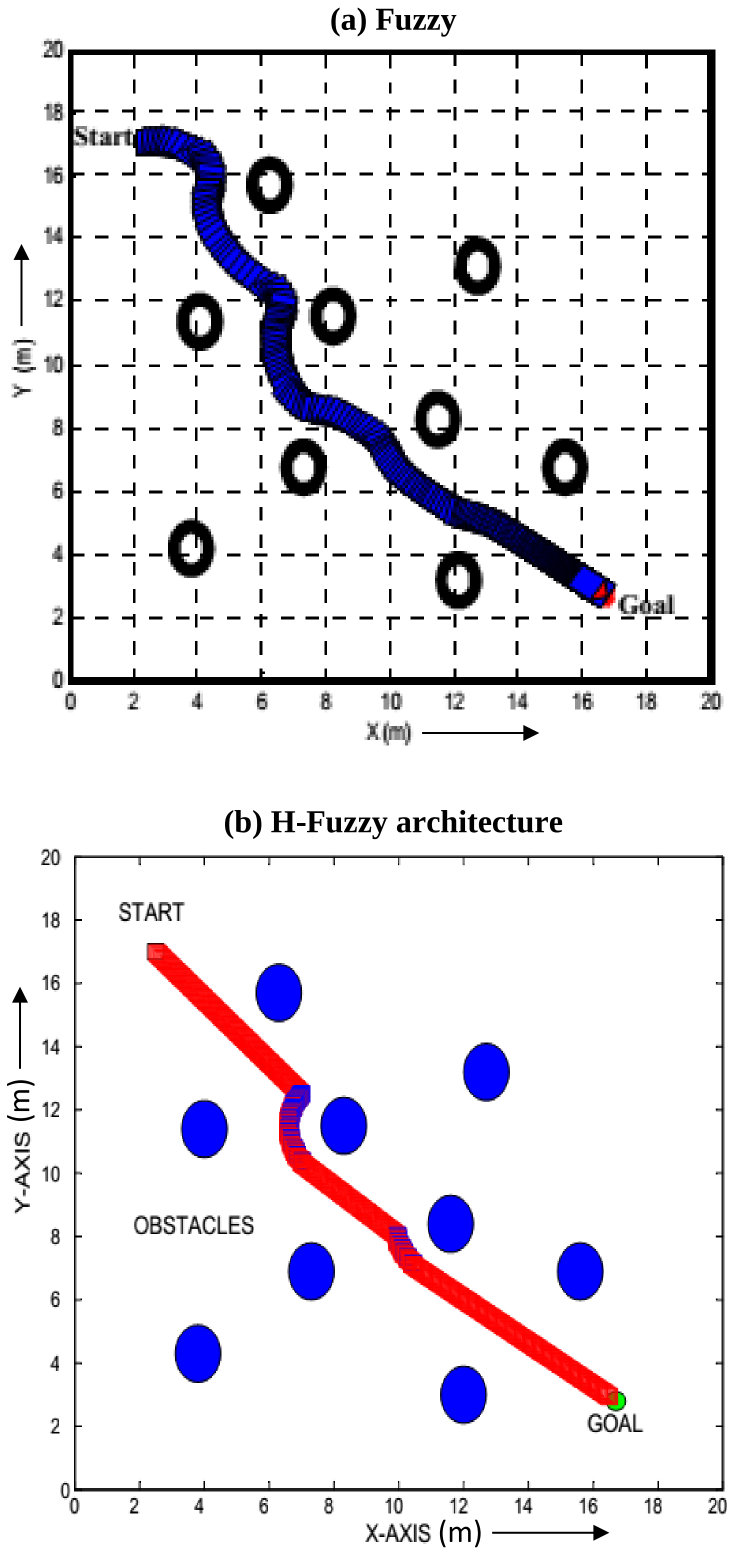


Figure 4.13: A simulation comparison results between **(a)** Fuzzy [163] and **(b)** H-Fuzzy architecture.

(a) ANN

(b) H-Fuzzy architecture

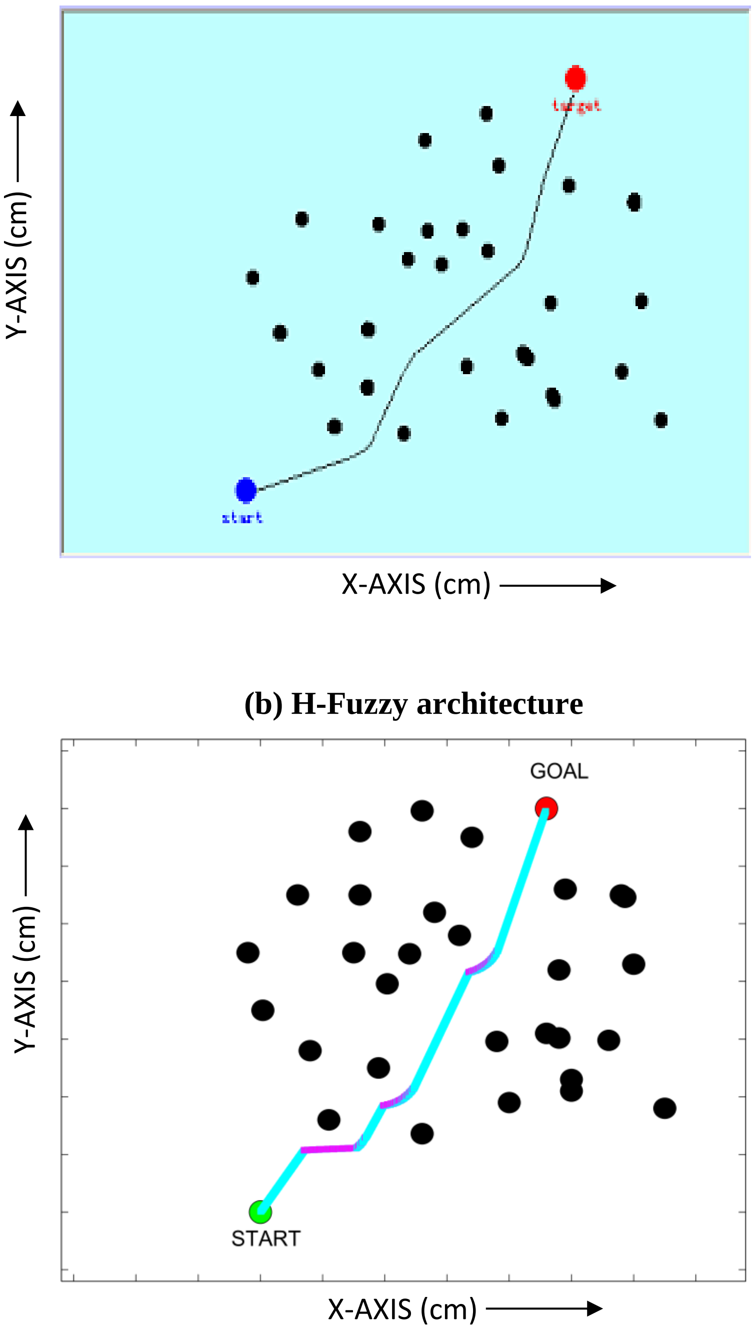


Figure 4.14: A simulation comparison results between **(a)** ANN [53] and **(b)** H-Fuzzy architecture.

Table 4.4: Comparison of simulation results between Fuzzy [163] method over proposed H-Fuzzy architecture

Figure no.	**Method**	**Navigation path length (cm)**
Figure 4.13 (a)	Fuzzy [163]	91
Figure 4.13 (b)	H-Fuzzy architecture	84

Table 4.5: The simulation results of ANN [53] method over proposed H-Fuzzy architecture in the cluttered environment

Figure no.	**Method**	**Navigation path length (cm)**
Figure 4.14 (a)	ANN [53]	74
Figure 4.14 (b)	H-Fuzzy architecture	69

4.5 Experimental Studies

4.5.1 Arduino Microcontroller based Wheeled Mobile Robot Description

This section presents the description of the Arduino microcontroller based two-wheeled differential drive experimental mobile robot (see the Figure 4.15). The mobile robot has two wheels, which is connected to two separate DC geared motors. The motion and orientation of the robot are controlled by two independent DC geared motors, which provides the necessary torque to all the driving wheels. The width of the robot plate is 23cm, and the track width and height of robot are 30cm and 8cm, respectively. The mobile robot is equipped with one sharp infrared range sensor on the front side, and the two ultrasonic range finder sensors fitted on the left and right side of the robot, as shown in Figure 4.16. The sharp infrared range sensor reads obstacles up to 150cm, and the ultrasonic sensor reads obstacles from 2cm to 4m approximately. In this study, the minimum and maximum velocities of the wheeled mobile robot are between the 6.7-16.7cm/sec.

4.5.2 Experiments

The experiments are conducted using the Arduino microcontroller based mobile robot in unknown environments. Figures 4.17 and 4.18 show the real time motion and orientation of the experimental mobile robot in the two different environments. The width and height of the platform are 250cm and 250cm, respectively. In the experimental results, it is assumed that the position of the start point and goal point are known, but the positions of all the obstacles in the environment are unknown for the robot. The proposed architecture receives input (obstacle distances) from the front, left, and the right group of the sensors (Figure 4.16) to control the turning angle and motor velocities of the mobile robot during navigation. If the left obstacle is near to the mobile robot, then the robot turns right, i.e. the velocity of the right motor is less than the velocity of left motor. Similarly, if the right obstacle is near to the mobile robot, then the robot turns left, i.e. the velocity of the right motor is more than the velocity of left motor.

Table 4.6 shows the experimental path length and time taken by the mobile robot to reach the goal using H-Fuzzy architecture in two different environments. Tables 4.7 and 4.8 illustrate the travelling path length and the navigation time comparison between the simulation and experiments. In the comparison study between the simulation and experiments, it is observed that some errors have been found, these are happened due to slippage and friction during real time experiment.

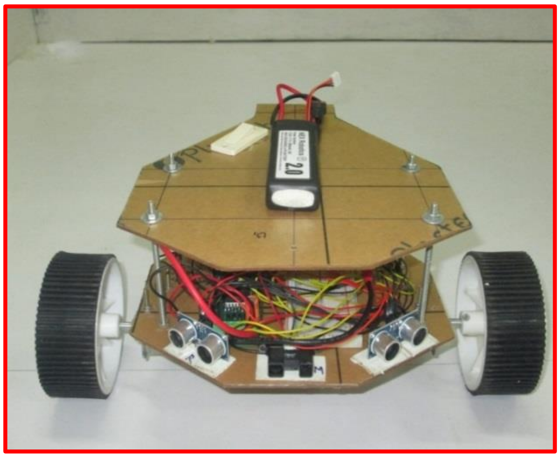


Figure 4.15: Arduino microcontroller based experimental mobile robot.

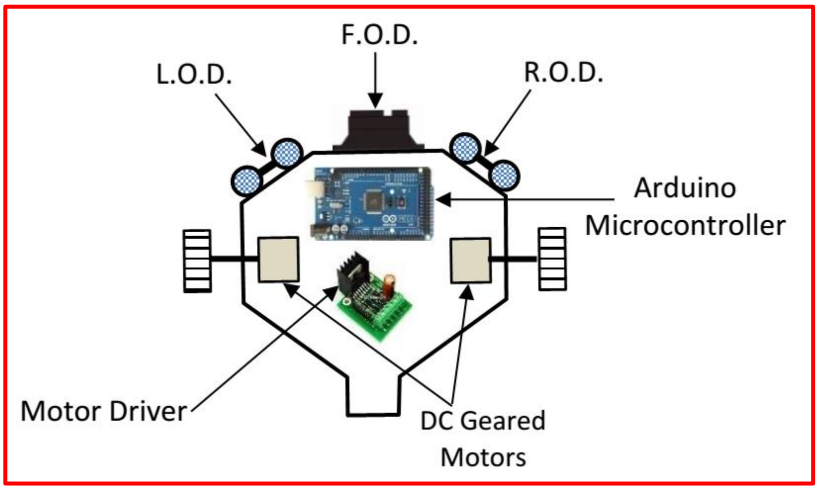


Figure 4.16: Sensor distribution of the experimental mobile robot.

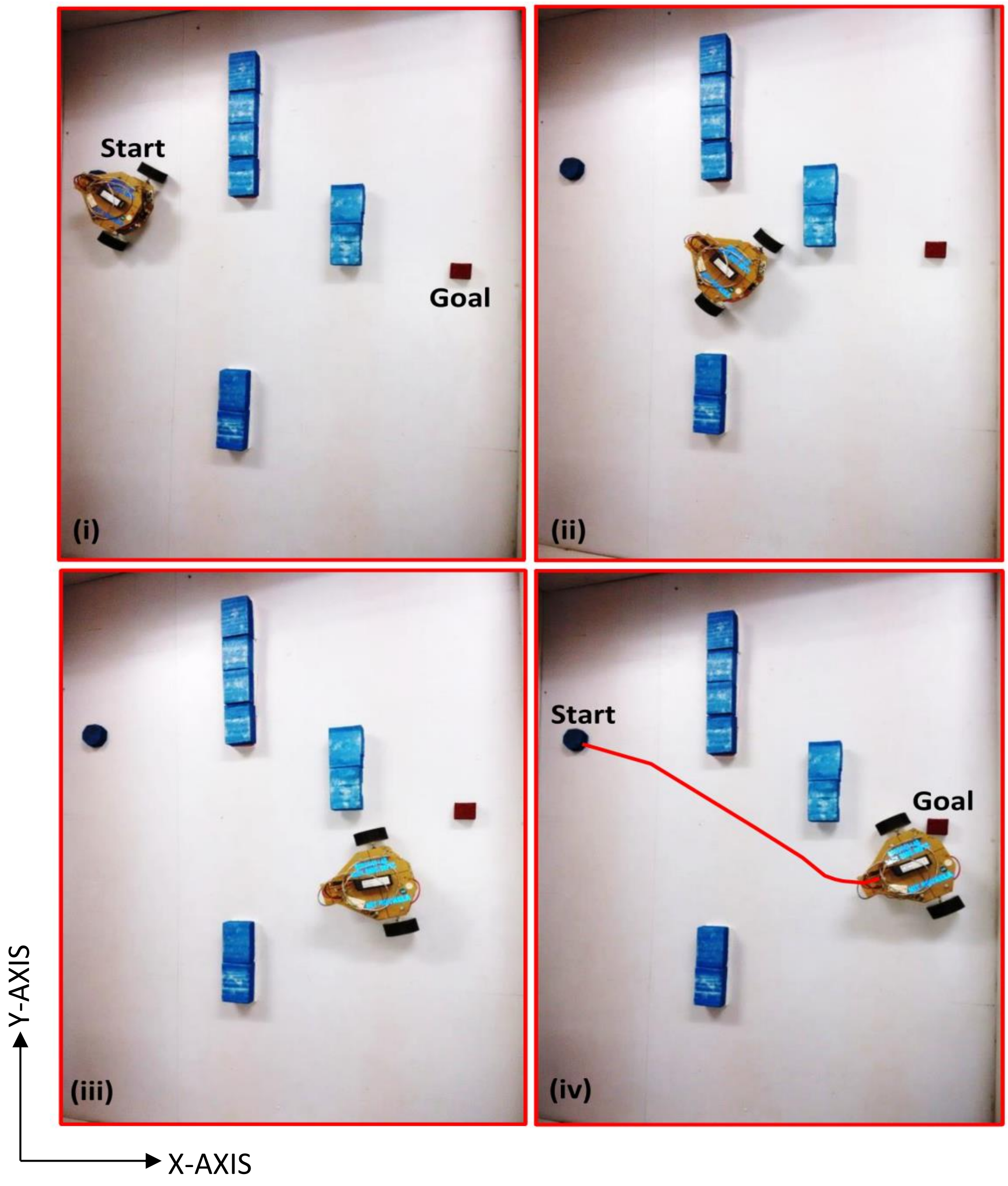


Figure 4.17: Experimental result of mobile robot navigation same as a simulation result (shown in Figure 4.9).

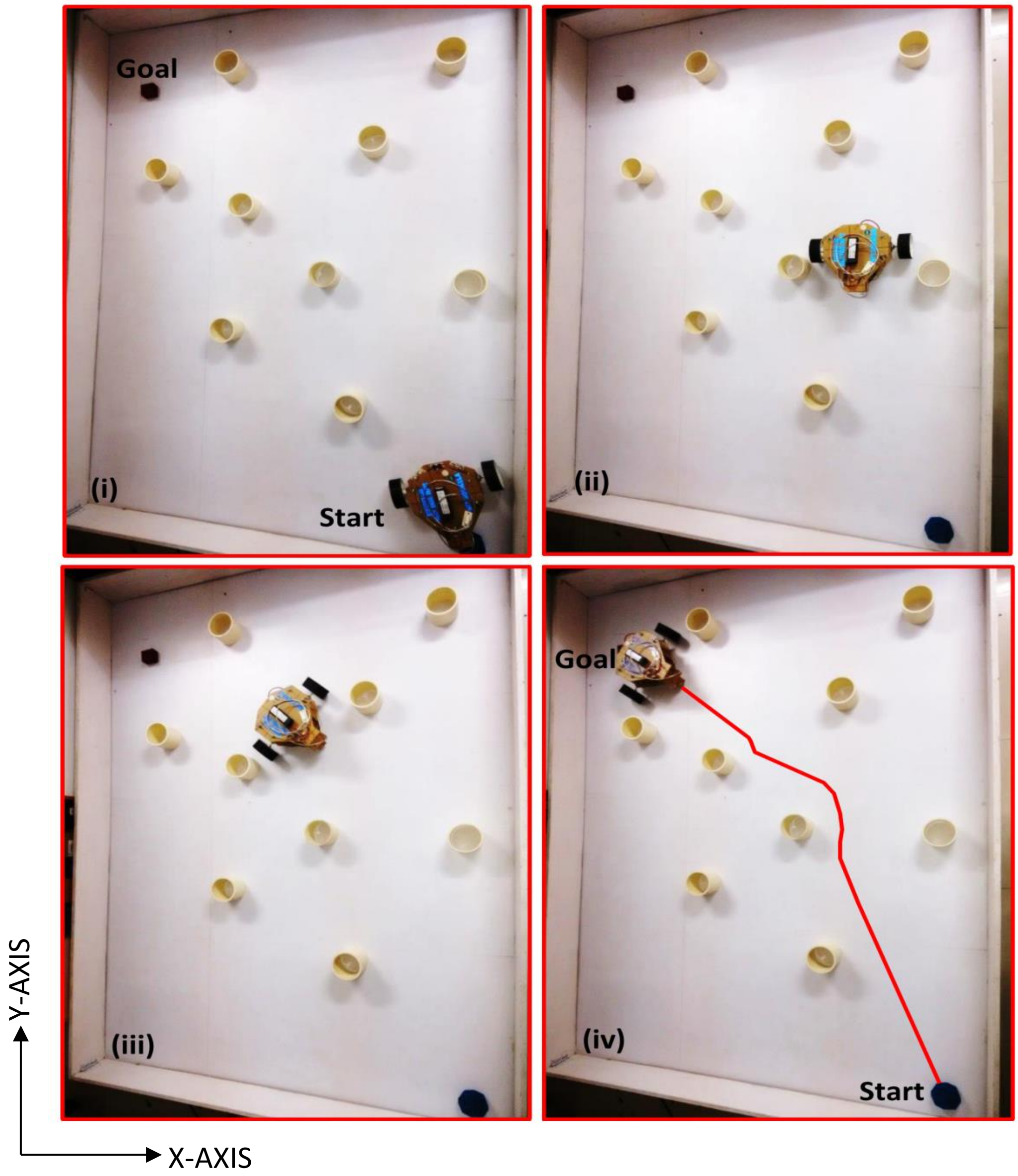


Figure 4.18: Experimental result of mobile robot navigation same as a simulation result (shown in Figure 4.13 (b)).

Table 4.6: Experimental results of mobile robot navigation in the different static and dynamic environments using H-Fuzzy architecture

Figure no.	Environment type	Travelling path length (cm)	Navigation time (sec)
Figure 4.17	Unknown environment	78	9.9
Figure 4.18	Unknown environment	90	11.6

Table 4.7: Travelling path lengths comparison between simulation and experimental results

Figure no. (Simulation and experimental res.)	Travelling path length (cm)		Error between simulation and experimental result
	Simulation result	Experimental result	
Figures 4.9 and 4.17	73	78	6.41%
Figures 4.13 (b) and 4.18	84	90	6.67%

Table 4.8: Navigation time comparison between simulation and experimental results

Figure no. (Simulation and experimental res.)	Navigation time (sec)		Error between simulation and experimental result
	Simulation result	Experimental result	
Figures 4.9 and 4.17	9.2	9.9	7.07%
Figures 4.13 (b) and 4.18	10.8	11.6	6.89%

4.6 Summary

This chapter introduces the hybrid fuzzy (H-Fuzzy) architecture for intelligent navigation of a mobile robot. The proposed H-Fuzzy architecture has been designed for two behaviors, i.e. goal reaching and obstacle avoidance.

Takagi-Sugeno type fuzzy logic architecture (TFa) has been used to assist the robot to reach the goal. Mamdani-type fuzzy logic architecture (MFa) has been applied to control the right motor velocity and left motor velocity of the mobile robot.

Furthermore, the proposed architecture generates better results (in terms of path length) as compared to previous models [163] and [53], which verifies the superiority of the proposed architecture.

Moreover, the simulation and experimental studies demonstrate that the proposed architecture efficiently drives the mobile robot in the different static and dynamic environments such as unknown, indoor and complex environments. In the comparison study between the simulation and experiment results errors are recorded, and the errors are found due to the effect of slippage and friction between the wheels of the robot and surface during navigation in real time mode.

During experiment utmost care has been taken to minimize the slippage and friction between the wheels and surface. Still the effect of slippage and friction are unavoidable, and errors are recorded during the comparison of the results for travelling path length (6.54%) and for navigation time (6.98%).

Chapter 5

Intelligent Navigation Control of a Mobile Robot in Unknown Environments using Cascade Neuro-Fuzzy Architecture

5.1 Introduction

Real-time navigation in the partially unknown environment is an interesting task for mobile robotics. This chapter presents the cascade neuro-fuzzy (CN-Fuzzy) architecture for intelligent navigation control of a mobile robot in an unknown environment filled with obstacles. The array of ultrasonic range finder sensors and sharp infrared range sensors are used to read the front, left and right obstacle distances. The cascade neural network is used to train the robot to reach the goal. Its inputs are the different obstacle distance received from the sensors. The output of the neural network is a turning angle between the robot and goal. The fuzzy architecture is integrated with the cascade neural network to control the velocities of the robot. Successful simulation and experimental results verify the effectiveness of the proposed architecture in both static and dynamic environments.

The applications of the intelligent robot in many fields such as industry, space, agriculture, defence and transportation, and other social sectors are growing day by day. The mobile robot performs many tasks such as rescue operation, patrolling, underwater exploration, disaster relief and planetary exploration, etc. Therefore, the author is trying to put the effort in the field of the intelligent robot using CN-Fuzzy architecture, which can avoid the obstacle autonomously and reach the goal safely in the given environment. Autonomous mobile robot navigation is one of the challenging tasks for any soft computing techniques. Fuzzy logic and neural network have been widely used for mobile robot navigation and control because these methods are capable of handling the system uncertainty. Generally, the fuzzy logic is the combination of fuzzy rules and membership functions (inputs and outputs), which are constructed by human knowledge. The neural network with fuzzy logic [77] improves the decision speed of the mobile robot for target seeking and obstacle avoidance.

Cascade neural network (CNN) is similar to feed forward neural network (FNN). Both neural networks use back propagation algorithm for updating the weights and biases [164]. This chapter describes the cascade neural network based fuzzy architecture for mobile-robot navigation in unknown environments. The cascade neural network is used to train the robot to reach the goal. Its inputs are different obstacle distance received from the sensors. The output of the neural network is a turning angle between the robot and goal. The fuzzy logic architecture is used to control the right motor velocity and left motor velocity of the mobile robot. In the last two decades, many researchers have implemented different neuro-fuzzy techniques for solving the navigation problem of the mobile robot. Motivated by the above literature survey, the primary objective of this chapter is to improve the path planning accuracy and efficiency of the mobile robot using the cascade neuro-fuzzy controller. The remainder of this chapter is structured as follows: Section 5.2 introduces the design and implementation of the CN-Fuzzy architecture for navigation of mobile robot and obstacle avoidance in unknown environments. Section 5.3 demonstrates the computer simulation results in different unknown environments. Section 5.4 describes the simulation result comparison with previous works. Section 5.5 presents the experimental results and discussion for validating the proposed controller. Finally, Section 5.6 depicts the summary.

5.2 Cascade Neuro-Fuzzy (CN-Fuzzy) Architecture

This section introduces the design and implementation of the CN-Fuzzy architecture for navigation of mobile robot and obstacle avoidance in unknown environments. The cascade neural network is used to train the robot to reach the goal in the environment, and the fuzzy logic architecture is used to control the right motor velocity and left motor velocity of the mobile robot. Figure 5.1 shows the proposed architecture of CN-Fuzzy for navigation of mobile robot and obstacle avoidance in unknown environments.

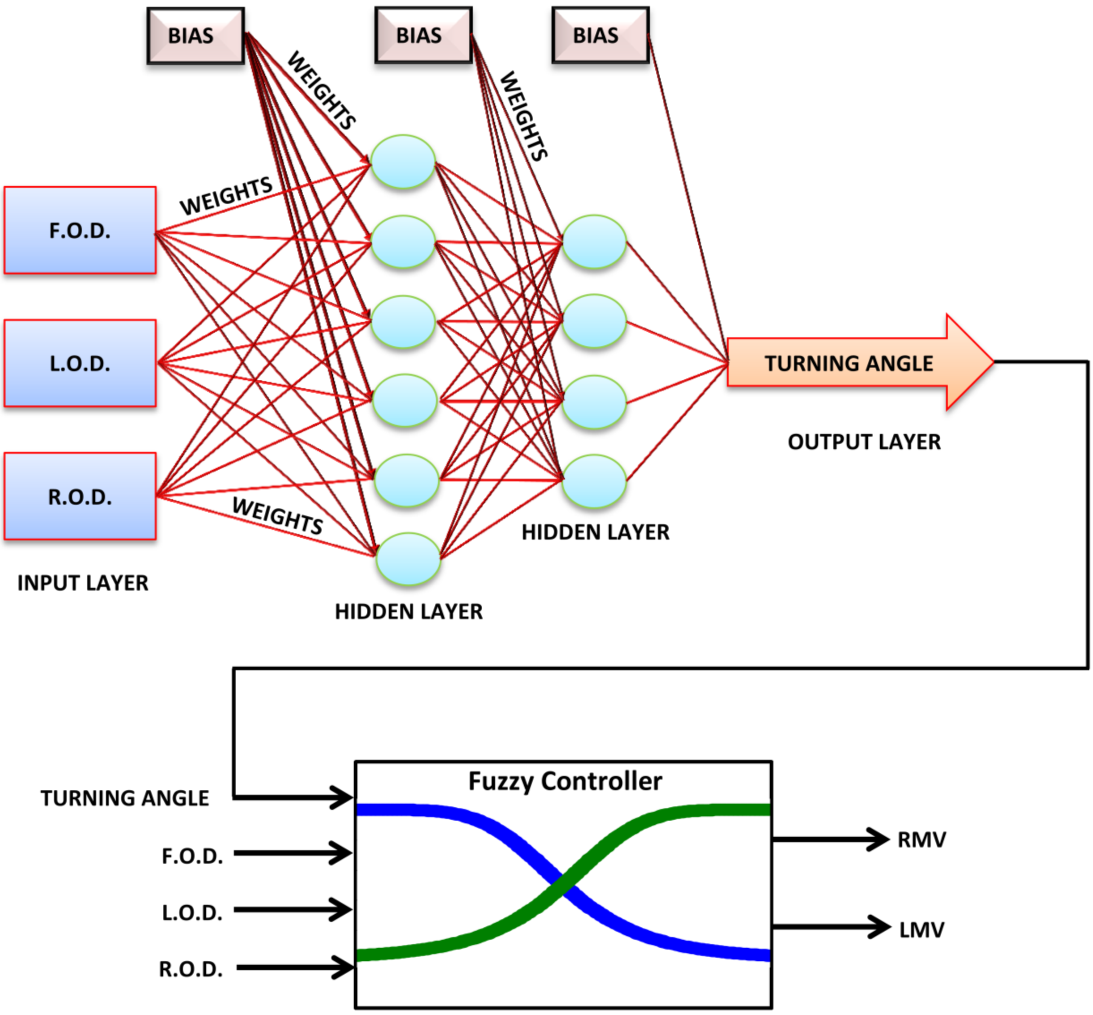


Figure 5.1: The cascade neuro-fuzzy architecture for navigation of mobile robot and obstacle avoidance in unknown environments.

5.2.1 Cascade Neural Network for Goal Reaching

The neural network is one of the important technique for the mobile robot navigation. In this section, the cascade neural network (CNN) is used to train the robot to reach the goal in the environment. The neural network is the combination of many layers such as input layer, hidden (intermediate) layers, and the output layer; all the layers are connected with each other by the neurons. The CNN is the similar to the feed forward neural network (FNN). Both CNN and FNN use back propagation algorithm for updating the weights and biases. The two back propagation algorithms, namely Levenberg-Marquardt (LM) and Bayesian regularization (BR) are used to adjust the network weights and biases. Figure

5.2 illustrates the general structure of a cascade neural network (CNN). In Figure 5.2, u, w, b, and v addresses the input variables, synaptic weights, neuron bias, and output variable, respectively.

The inputs of the CNN are the obstacle distance received from the various sensors, and the output of the CNN is a turning angle between the robot and goal. Table 5.1 describes the different training patterns for the cascade neural network, which helps the robot to reach the goal in the environment. The proposed CNN uses three inputs, two hidden layers (six and four neurons, respectively) and single output layer for the mobile robot navigation. The CNN has three inputs: F.O.D. (Front Obstacle Distance), L.O.D. (Left Obstacle Distance), and R.O.D. (Right Obstacle Distance), respectively. The output of this CNN is a turning angle (T.A.) between the robot and goal. The input and output of the CNN can be written as follows: -

Input layer (first layer):

$$q_i^{[1]} = u_i \tag{5.1}$$

where i = 1, 2, 3. (Three inputs F.O.D., L.O.D., and R.O.D., respectively)

Two hidden layers (second and third):

$$q_t^{[s]} = \varphi\left(NET_t^{[s]}\right) \tag{5.2}$$

$$NET_t^{[s]} = \sum_i (w_{t \cdot i}^{[s]} \cdot q_i^{[s-1]} + b_t^{[s]}) \tag{5.3}$$

where s = 2, 3. (Second and third layers)

Output layer (fourth layer):

$$v_{(p)} = q^{[4]} = \varphi\left(NET^{[4]}\right) \tag{5.4}$$

$$NET^{[4]} = \sum_i (w_i^{[4]} \cdot q_i^{[3]} + b_i^{[4]}) \tag{5.5}$$

where u_i is the input variables, $v_{(p)}$ is the predicted output variable (turning angle). The $w_{t \cdot i}^{[s]}$ is the synaptic weight on connection joining the i th neuron in the layer $[s-1]$ to the

t th neuron in the layer $[s]$; $b_t^{[s]}$ is a bias of the t th neuron in the layer $[s]$, and $\varphi_t(d)$ is the Log-sigmoid transfer function.

$$\varphi(d) = \frac{1}{\left(1+\exp(-d)\right)} \tag{5.6}$$

The proposed CNN is verified through the mean squared error (MSE) and root mean square error (RMSE) method: -

$$MSE(\%) = \left[\sum\nolimits_1^r \left(\frac{v_{(a)} - v_{(p)}}{r}\right)^2\right] \times 100 \tag{5.7}$$

$$RMSE(\%) = \sqrt{\frac{1}{r}\left[\sum\nolimits_1^r \left(\frac{v_{(a)} - v_{(p)}}{v_{(a)}}\right)^2\right]} \times 100 \tag{5.8}$$

where $v_{(a)}$ is the actual output variable, $v_{(p)}$ is the predicted (network) output variable, and r is the number of observations.

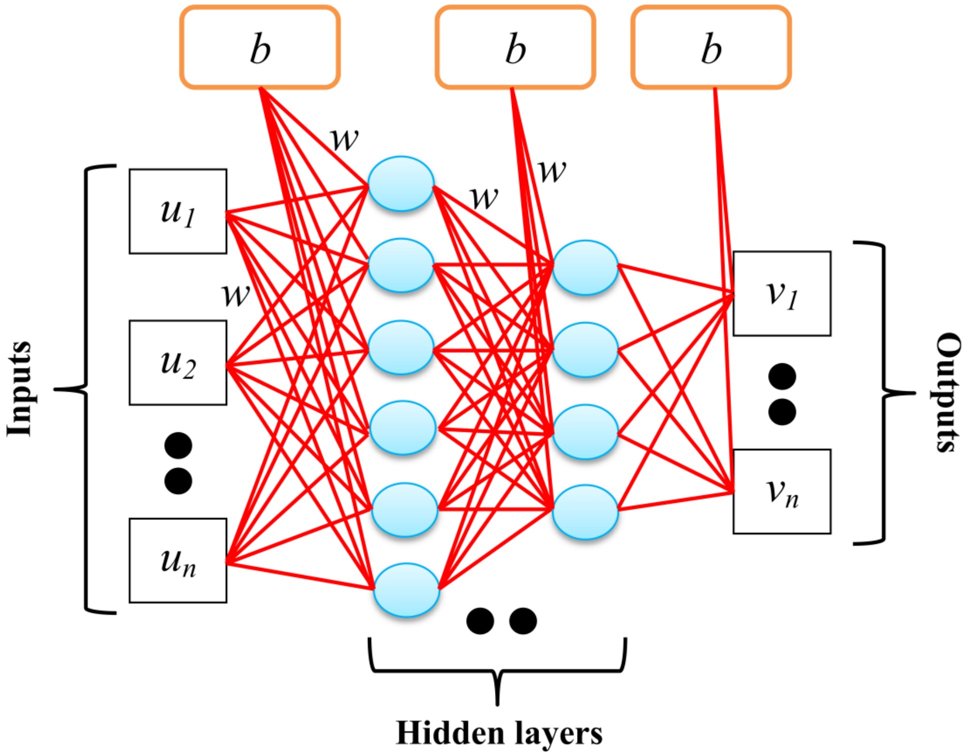


Figure 5.2: The general structure of the cascade neural network (CNN).

Table 5.1: The different training patterns for mobile robot navigation

F.O.D. (cm)	L.O.D. (cm)	R.O.D. (cm)	T.A. (degree)	Turning direction
20	115	20	74.3	Left
20	20	150	-65.9	Right
125	25	150	-70.4	Right
25	75	50	55	Left
40	120	60	59.4	Left
25	150	100	72.8	Left
25	50	120	-22.9	Right
22	25	22	73.4	Left
50	25	25	0	Straight
20	27	27	77	Left
100	28	25	0	Straight
25	21	22	77.2	Left
150	25	115	-70.5	Right
150	20	25	0	Straight
150	100	100	-70.4	Right

5.2.2 Fuzzy logic architecture (FLA) for obstacle avoidance

This section describes the design of Mamdani-type fuzzy logic architecture for navigation of mobile robot and obstacle avoidance in unknown environments. The fuzzy logic architecture (FLA) is used to control the right motor velocity and left motor velocity of the mobile robot. The proposed FLA has four inputs and two outputs. The FLA receives first three inputs (obstacle distance) from the various sensors of the mobile robot. The first three inputs are denoted by F.O.D., L.O.D., and R.O.D., respectively. The fourth input is the turning angle (goal angle) between the robot and goal, and which is received from the CNN. The outputs of the FLA are the velocities of the motors of robot. The outputs are addressed by RMV (Right Motor velocity) and LMV (Left Motor velocity), respectively. The range of first three inputs is divided into two linguistic variables, namely CLOSE and AWAY, respectively, and it is located between 20cm to 150cm. The two linguistic variables NEGATIVE and POSITIVE, respectively, are used for turning angle. The range of outputs is divided into two linguistic variables, namely LOW and HIGH, respectively. The two generalized bell-shaped (Gbell) membership functions are used for inputs and outputs. Figure 5.3 shows the input and output variables of the FLA. Figure 5.4 illustrates the fuzzy logic architecture. The fuzzy rule set of the FLA is described in Table 5.2. The FLA is composed through Mamdani-type fuzzy model in the following form: -

$$Rule_m : IF\ x_1\ is\ A_{j1},\ x_2\ is\ A_{j2},\ x_3\ is\ A_{j3},\ \&\ x_4\ is\ A_{j4}\ THEN\ y_1\ is\ B_{j1}\ \&\ y_2\ is\ B_{j2} \tag{5.9}$$

where m=1, 2, 3…12 (twelve rules), the x_1, x_2, x_3, and x_4 are the input variables. Similarly, y_1 and y_2 are the output variables. The A_{j1}, A_{j2}, A_{j3}, and A_{j4} are the fuzzy sets of the input variables. Similarly, B_{j1} and B_{j2} are the fuzzy sets of the output variables. The j=1, 2, because each input and output have two Gbell membership functions. The fuzzy set (inputs and outputs) uses the following Gbell membership function: -

$$\mu_{jk}(x_k; a, b, c) = \frac{1}{1 + \left| \frac{x_k - c_{jk}}{a_{jk}} \right|^{2b_{jk}}} \tag{5.10}$$

$$\mu_{jl}(y_l; a, b, c) = \frac{1}{1 + \left| \frac{y_l - c_{jl}}{a_{jl}} \right|^{2b_{jl}}} \tag{5.11}$$

where k=1…4 (four inputs), and l=1, 2 (two outputs). The symbols a, b, and c are adjusting parameters of the Gbell membership function; called as the half width, slope control, and center respectively.

The defuzzification of the output variables (y_1 and y_2) are accomplished by the weighted average method: -

$$y_1 = \frac{\sum_{m=1}^{12} \left(\mu_{j1}(x_1) \cdot \mu_{j2}(x_2) \cdot \mu_{j3}(x_3) \cdot \mu_{j4}(x_4) \right) \cdot y_1}{\sum_{m=1}^{12} \left(\mu_{j1}(x_1) \cdot \mu_{j2}(x_2) \cdot \mu_{j3}(x_3) \cdot \mu_{j4}(x_4) \right)} \tag{5.12}$$

$$y_2 = \frac{\sum_{m=1}^{12} \left(\mu_{j1}(x_1) \cdot \mu_{j2}(x_2) \cdot \mu_{j3}(x_3) \cdot \mu_{j4}(x_4) \right) \cdot y_2}{\sum_{m=1}^{12} \left(\mu_{j1}(x_1) \cdot \mu_{j2}(x_2) \cdot \mu_{j3}(x_3) \cdot \mu_{j4}(x_4) \right)} \tag{5.13}$$

Table 5.2: Fuzzy rule sets for navigation of mobile robot and obstacle avoidance

Fuzzy rules	**F.O.D. (cm)**	**L.O.D. (cm)**	**R.O.D. (cm)**	**T. A. (degree)**	**RMV (cm/sec)**	**LMV (cm/sec)**
1	Away	Away	Away	Positive	High	Low
2	Away	Away	Away	Negative	Low	High
3	Close	Close	Close	Negative	Low	High
4	Close	Close	Close	Positive	High	Low
5	Away	Close	Away	Negative	Low	High
6	Away	Away	Close	Positive	High	Low
7	Close	Away	Away	Negative	Low	High
8	Close	Away	Away	Positive	High	Low
9	Close	Close	Away	Negative	Low	High
10	Close	Away	Close	Positive	High	Low
11	Away	Close	Close	Positive	High	Low
12	Away	Close	Close	Negative	Low	High

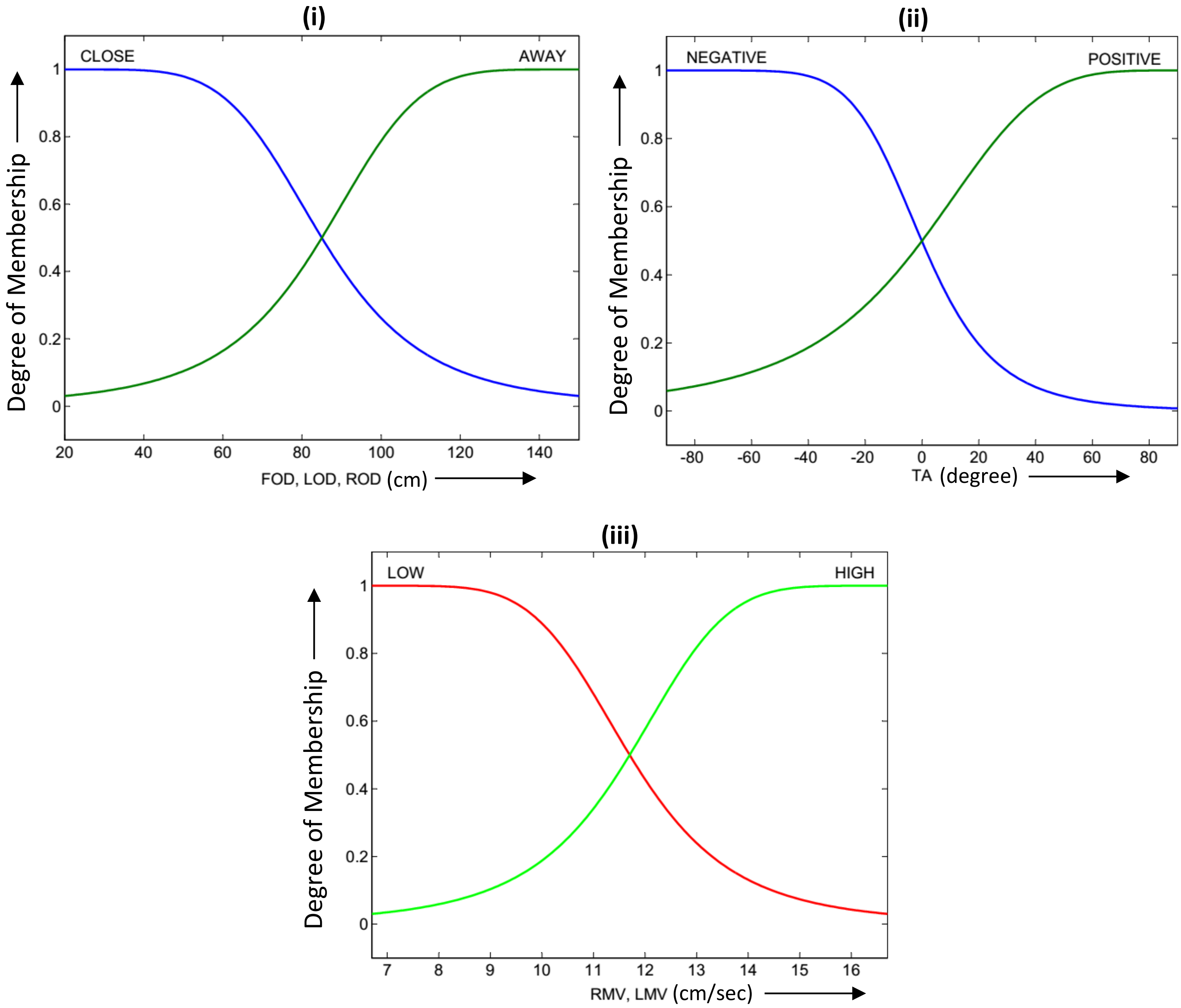


Figure 5.3: Membership functions **(i)** Obstacle distances (F.O.D., L.O.D. and R.O.D., respectively), **(ii)** Turning angle (TA), and **(iii)** Motor velocities (Right and Left respectively).

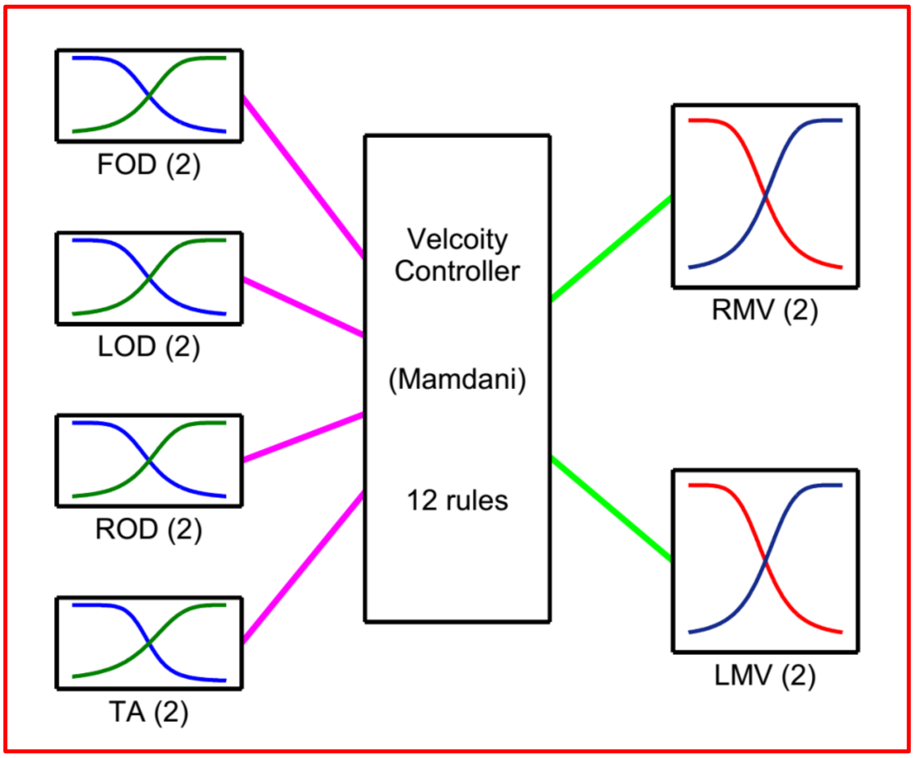


Figure 5.4: Fuzzy logic architecture.

5.3 Computer Simulation Results

This section presents the computer simulation results using CN-Fuzzy architecture in the different unknown static and dynamic environments. The simulations have done using MATLAB software on the HP 3.40 GHz processor. Figure 5.5 illustrates the developed flowchart of mobile robot navigation and obstacle avoidance based on CN-Fuzzy architecture. Figures 5.6 to 5.9 show the mobile robot navigation trajectories in the different static and dynamic environments. In the simulation results, it is assumed that the position of the start point and goal point are known. But the positions of all the obstacles in the environment are unknown for the robot. The dimension of the environments is 300cm width and 300cm height. A minimum threshold distance is fixed between the robot and the obstacle. Now if the robot detects the obstacle in the threshold range, then the proposed architecture estimates the desired turning direction of a mobile robot. Table 5.3 illustrates the navigation path length and time taken by the robot in the various unknown environments.

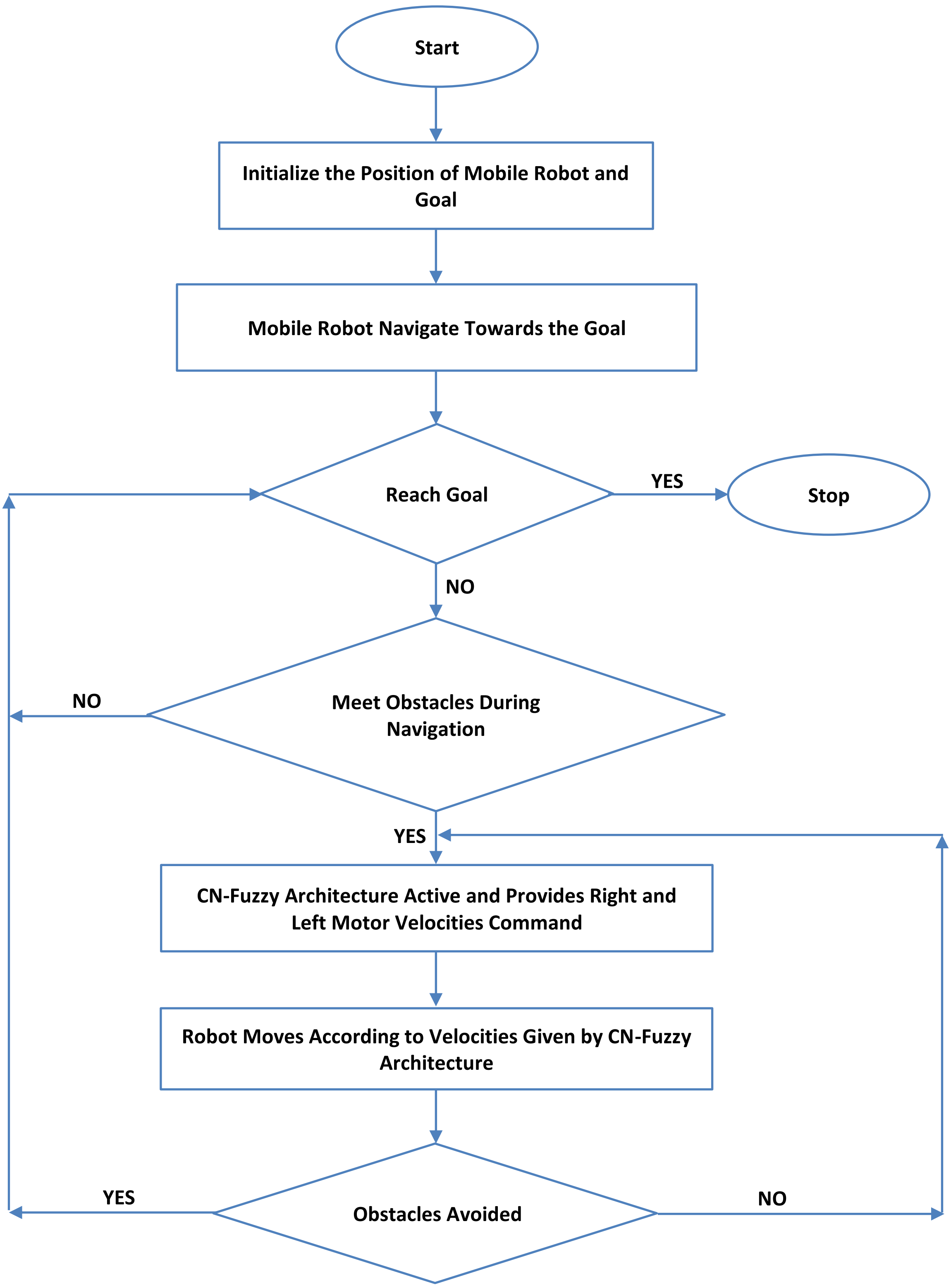


Figure 5.5: Flowchart of the mobile robot navigation and obstacle avoidance based on CN-Fuzzy architecture.

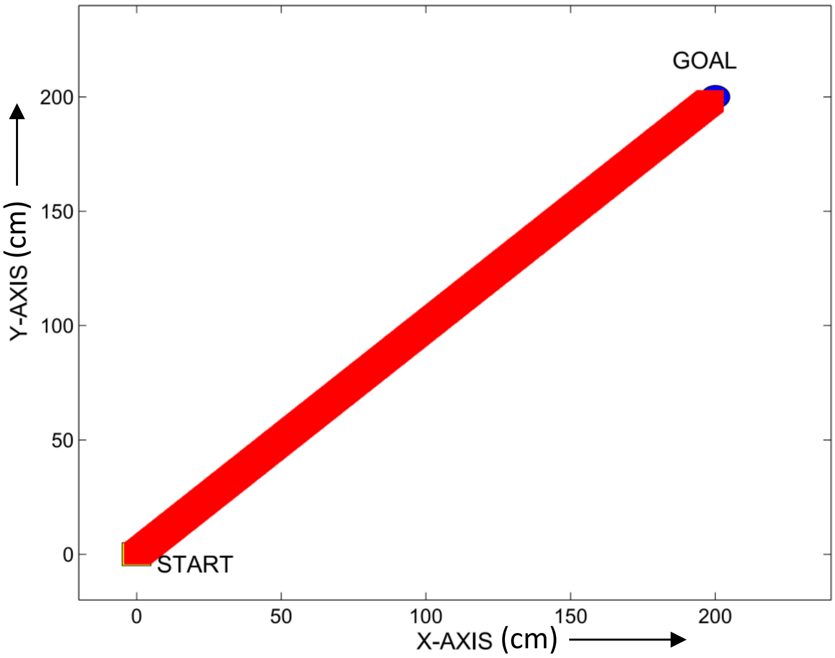


Figure 5.6: Mobile robot navigation in an environment without obstacle using CN-Fuzzy architecture.

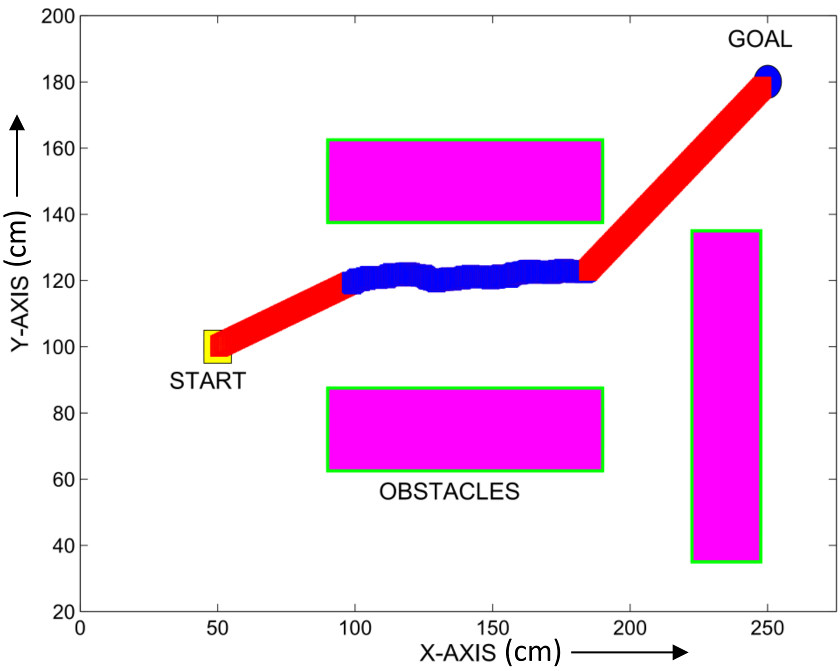


Figure 5.7: Mobile robot navigation in an unknown environment using CN-Fuzzy architecture.

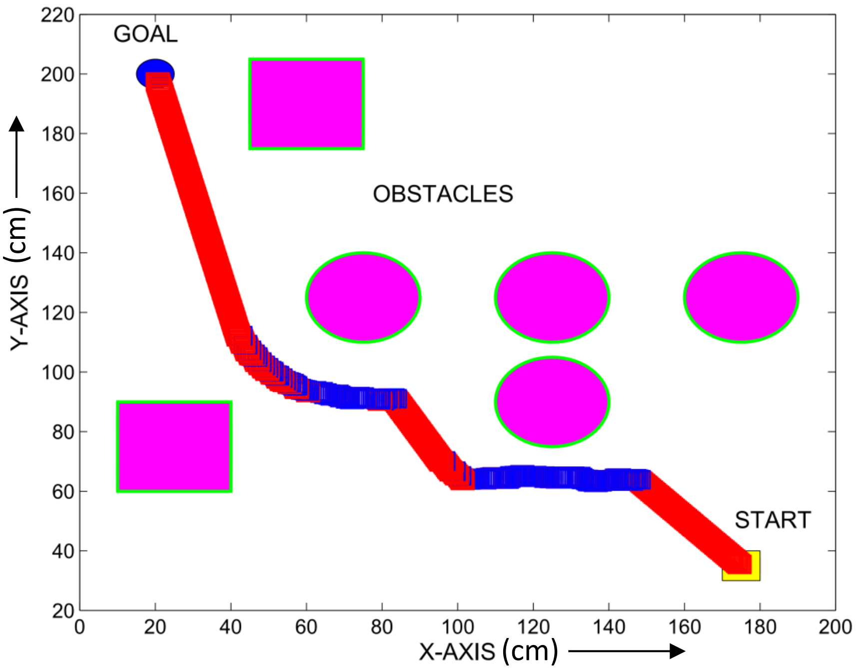


Figure 5.8: Mobile robot navigation in the cluttered environment using CN-Fuzzy architecture.

Table 5.3: Simulation results of mobile robot navigation in the different environments using CN-Fuzzy architecture

Figure no.	**Environment type**	**Travelling path length (cm)**	**Navigation time (sec)**
Figure 5.6	Without obstacle	103	11.6
Figure 5.7	Unknown environment	89	10.1
Figure 5.8	Cluttered environment	120	13.4
Figure 5.9	Dynamic environment	77	8.6

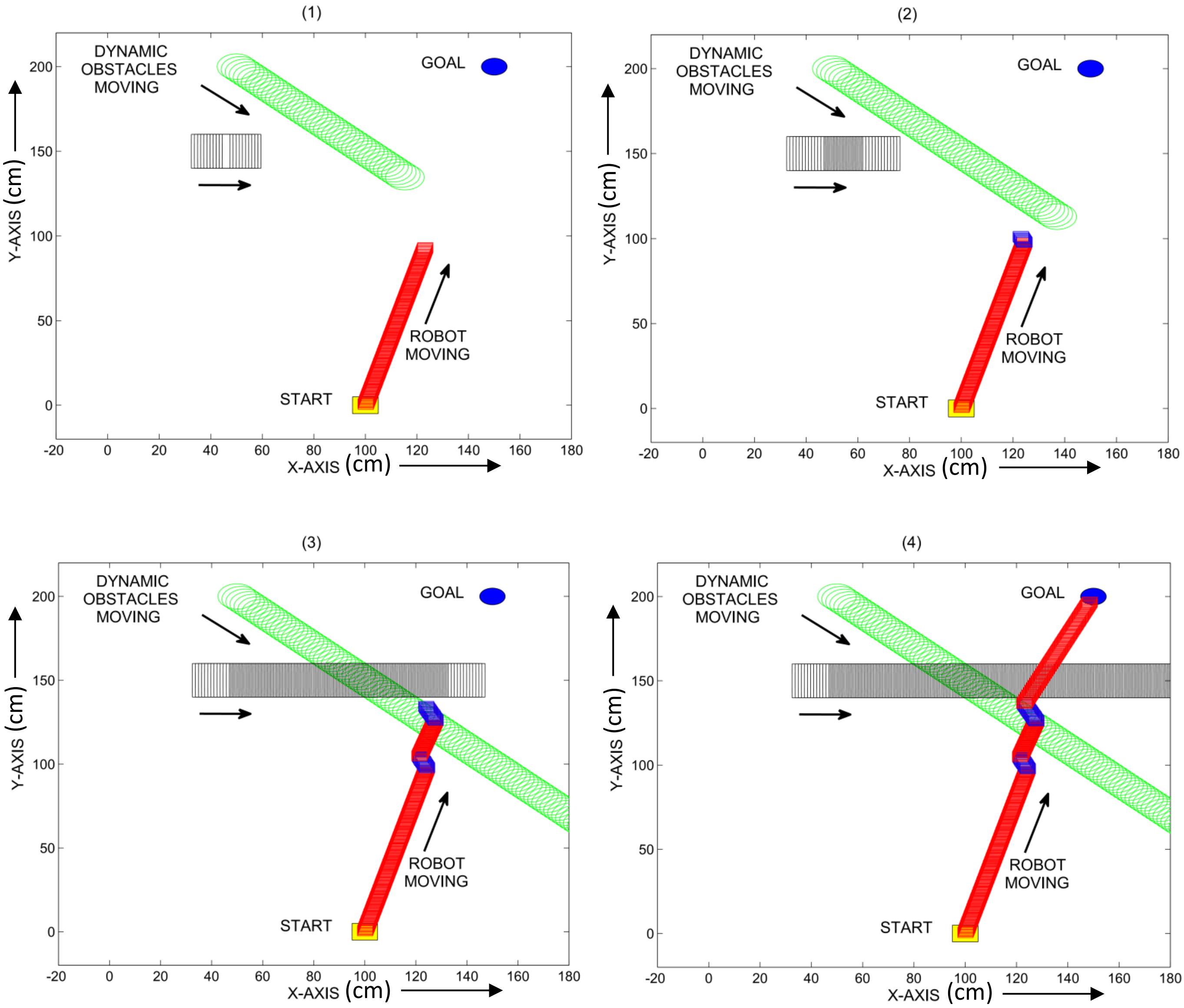


Figure 5.9: Mobile robot navigation in the dynamic environment using CN-Fuzzy architecture.

5.4 Comparison with Previous Works

This section describes the computer simulation result comparison between the previous models [28, 11] and proposed CN-Fuzzy architecture in the same environment.

5.4.1 First Comparison with Previous Works

In article [28], the authors have designed goal-seeking, obstacle avoidance behavior, and other behavior for mobile robot navigation by using fuzzy controller. Figures 5.10 and 5.11 illustrate the mobile robot navigation in the same environment without obstacle using fuzzy controller [28] and CN-Fuzzy architecture, respectively. From simulation result, it can be clearly seen that the robot covers shorter distance to reach the goal using proposed architecture as compared to previous model [28]. Table 5.4 shows the path covered by the robot to reach the goal using fuzzy controller [28] and proposed CN-Fuzzy architecture. The centimetre measurements are taken on the proportional basis.

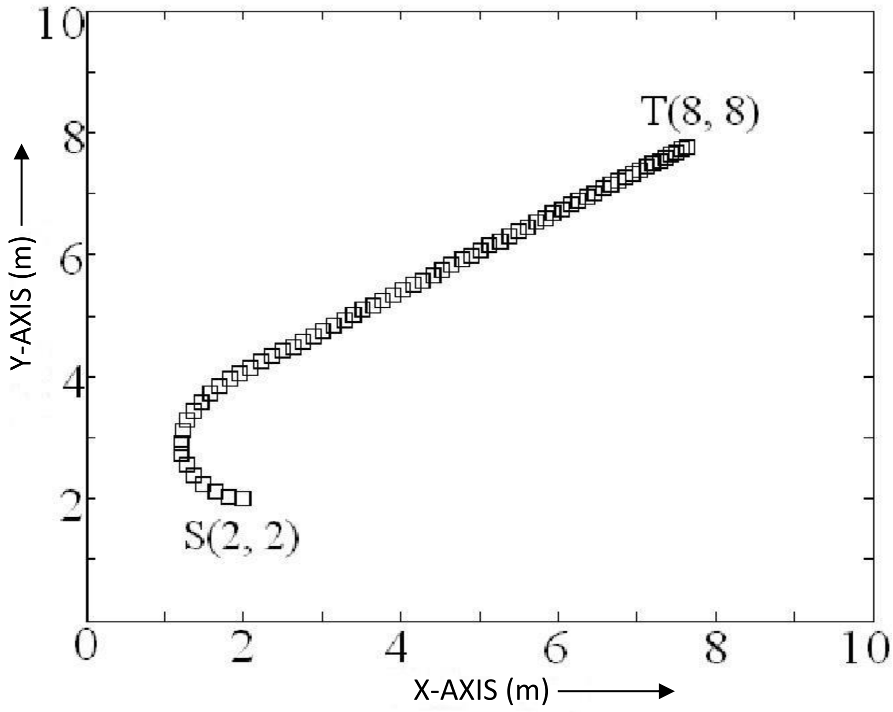


Figure 5.10: Mobile robot navigation in an environment without obstacle using fuzzy controller [28].

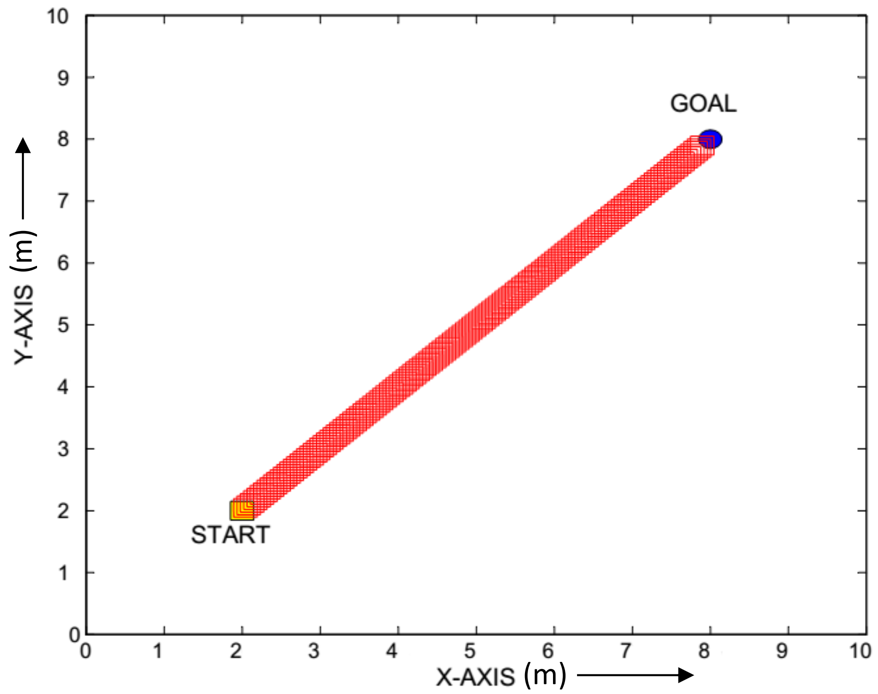


Figure 5.11: Mobile robot navigation in an environment without obstacle using CN-Fuzzy architecture.

Table 5.4: The simulation result comparison between the fuzzy controller [28] and proposed CN-Fuzzy architecture

Figure no.	**Method**	**Navigation path length (cm)**
Figure 5.10	Fuzzy [28]	51
Figure 5.11	CN-Fuzzy architecture	46

5.4.2 Second Comparison with Previous Works

In this section, the simulation result comparison has been made between the previous controller [11] and proposed CN-Fuzzy architecture in the same environment with the obstacles. In [11], the authors have discussed the motion and path planning of a car-like wheeled mobile robot between the stationary obstacles using backpropagation artificial neural network. Figure 5.12 shows the mobile robot navigation in an environment with obstacles using artificial neural network [11]. Figure 5.13 presents the path covered by the robot using proposed CN-Fuzzy architecture in the same environment. From the Figures 5.12 and 5.13, it is observed that the proposed architecture avoid the obstacles with some shorter distance or minimum steering as compared to previous model [11]. Table 5.5 illustrates the path traced (in cm) by the robot to reach the goal using proposed architecture and previous model [11]. The centimetre measurements are taken on the proportional basis.

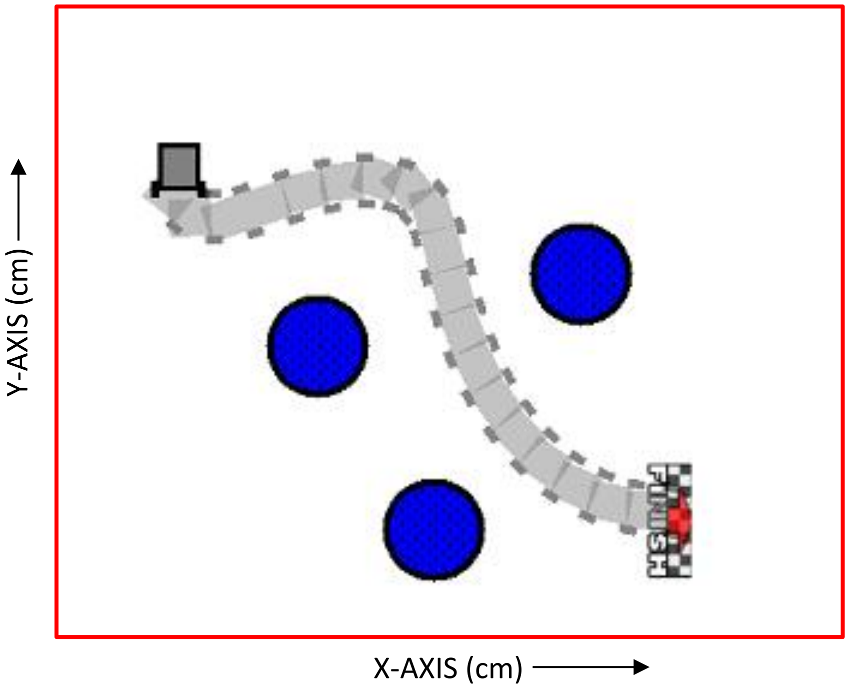


Figure 5.12: Mobile robot navigation in an environment with obstacles using artificial neural network [11].

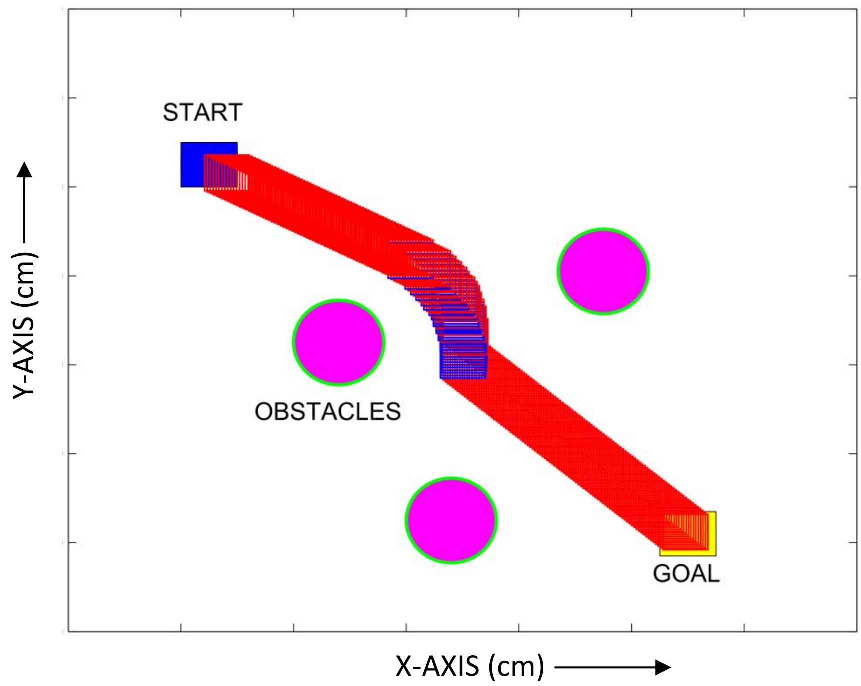


Figure 5.13: Mobile robot navigation in an environment with obstacles using CN-Fuzzy architecture.

Table 5.5: The simulation result comparison between the artificial neural network [11] and proposed CN-Fuzzy architecture

Figure no.	**Method**	**Navigation path length (cm)**
Figure 5.12	Artificial neural network [11]	87
Figure 5.13	CN-Fuzzy architecture	80

5.5 Experimental Results

5.5.1 Experimental Mobile Robot Description

This section describes the characteristic of the experimental mobile robot (Figure 5.14). The robot has two front wheels, which is powered by separate DC geared motors. The motor driver is used to control the velocity and direction of the robot. The width of the robot plate is 23cm, and the track width and height of robot are 30cm and 8cm, respectively. The mobile robot is equipped with one sharp infrared range sensor on the front side, and the two ultrasonic range finder sensors fitted on the left and right side of the robot, as shown in Figure 5.15. Each sensor can read obstacle from 20cm to 150cm approximately. The minimum and maximum velocities of the experimental mobile robot are between 6.7 cm/sec to 16.7 cm/sec approximately.

5.5.2 Experiments

This section presents the experimental results of a mobile robot using CN-Fuzzy architecture in the different environments. The experiments have been performed by C/C++ running Arduino microcontroller based mobile robot. The proposed architecture controls the motor velocities (right and left) of the robot during navigation in the environment using sensor data interpretation. Figures 5.16 to 5.18 show the real time navigation of the experimental mobile robot in the different environments. The width and height of the platform are 250cm and 250cm, respectively. In the experimental results, it is assumed that the position of the start point and goal point are known. But the positions of all the obstacles in the environment are unknown for the robot. Firstly, the robot goes towards the goal in the environment, and if the sensor detects the obstacle in the threshold range, then the proposed architecture controls the velocity of the mobile robot. The experimental results in the different snapshots verify the effectiveness of the proposed architecture. Table 5.6 shows the real-time navigation path length and time taken by the robot in the various unknown environments. Table 5.7 and Table 5.8 illustrates the travelling path length and navigation time comparison between the simulation and experimental results, respectively. In the comparison study between the simulation and experiments, it is observed that some errors have been found, these are happened due to slippage and friction during real time experiment.

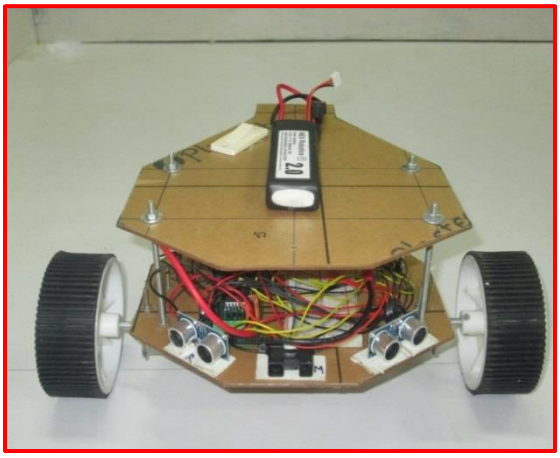


Figure 5.14: Experimental mobile robot.

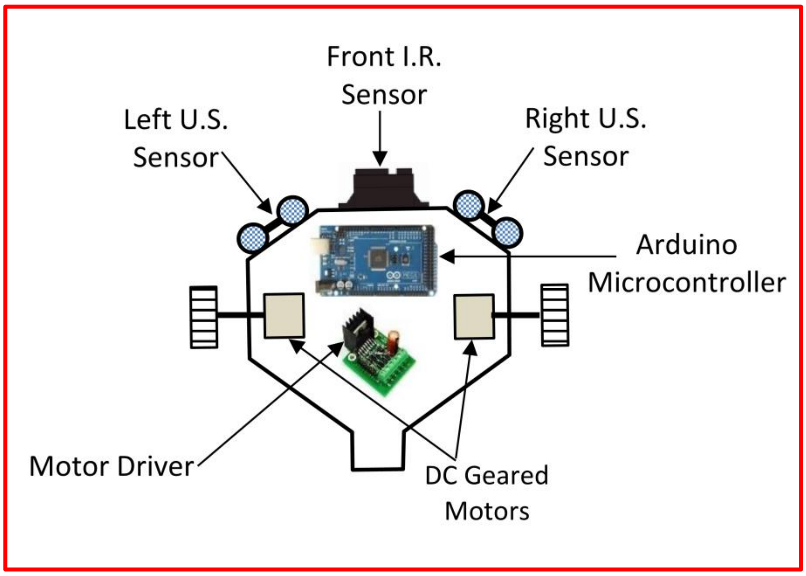


Figure 5.15: Sensor distribution of the experimental mobile robot.

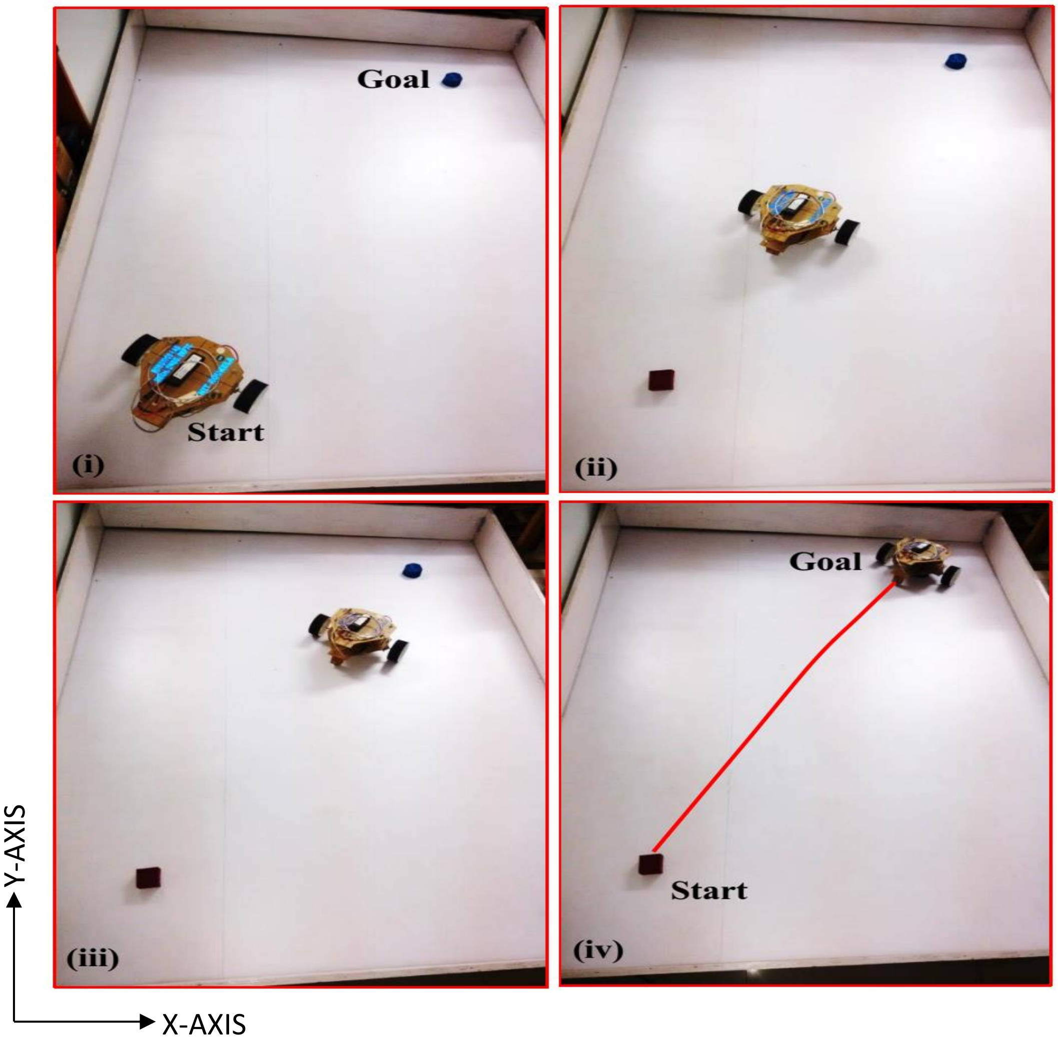


Figure 5.16: Experimental result of mobile robot navigation same as a simulation result (shown in Figure 5.6).

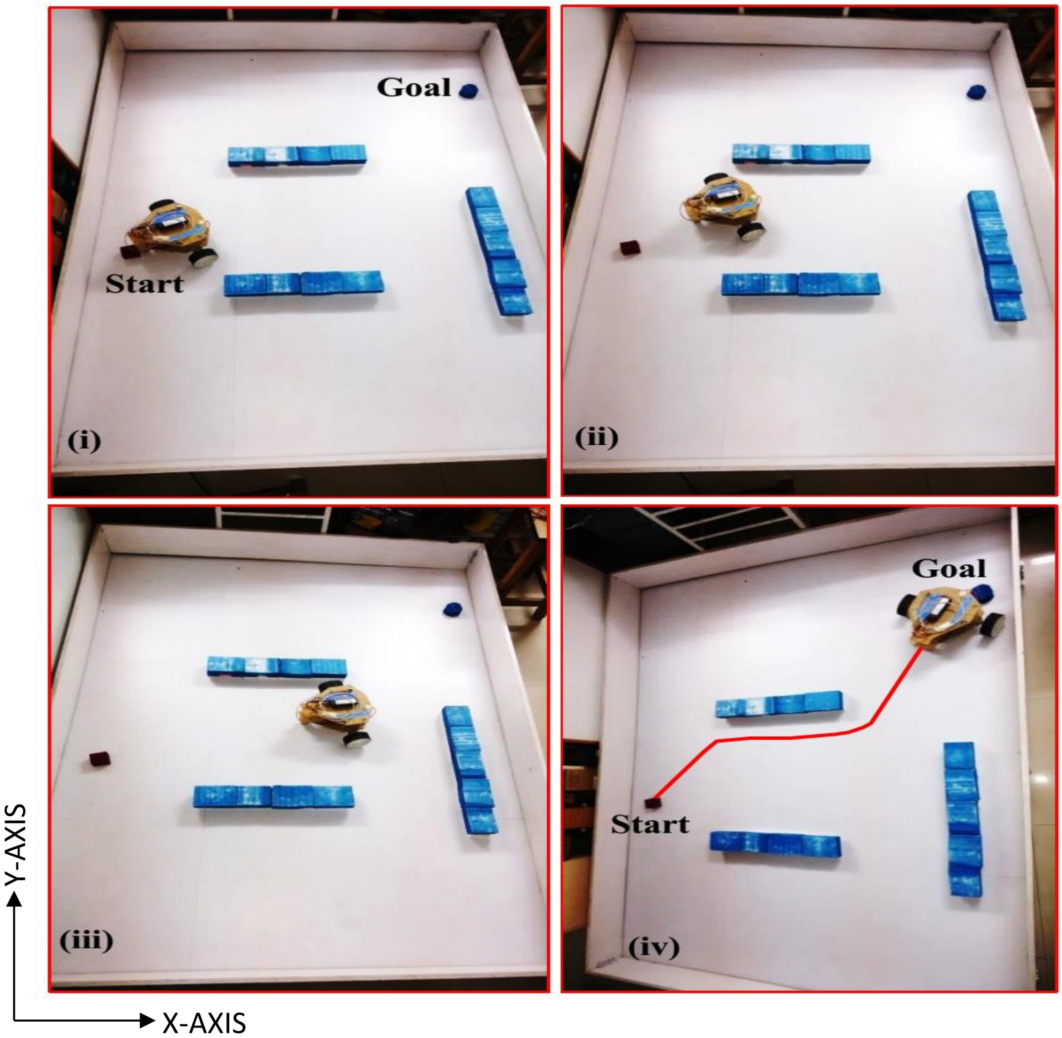


Figure 5.17: Experimental result of mobile robot navigation same as a simulation result (shown in Figure 5.7).

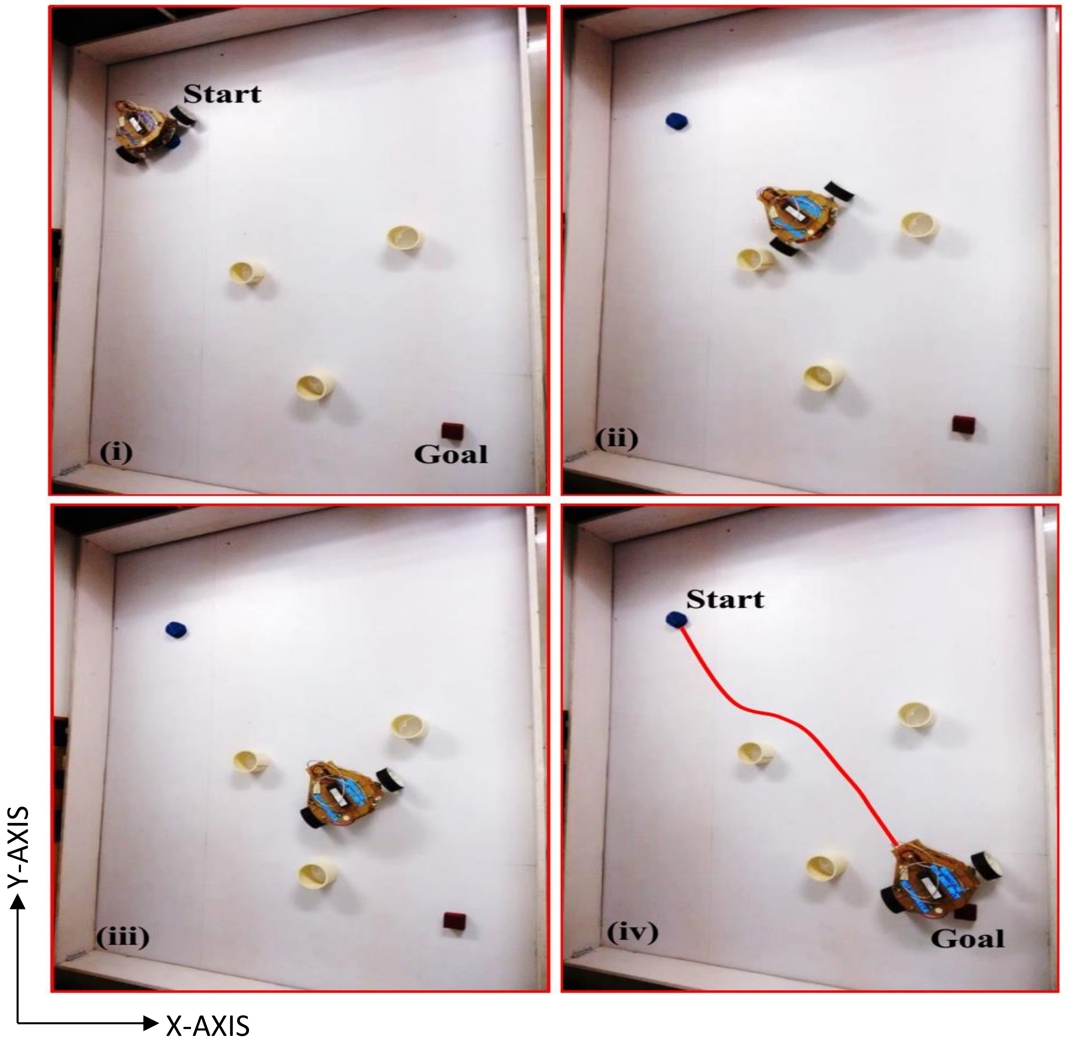


Figure 5.18: Experimental result of mobile robot navigation same as a simulation result (shown in Figure 5.13).

Table 5.6: Experimental results of a mobile robot navigation in the different environments using CN-Fuzzy architecture

Figure no.	**Environment type**	**Travelling path length (cm)**	**Navigation time (sec)**
Figure 5.16	Without obstacle	109	12.4
Figure 5.17	Unknown environment	94	10.8
Figure 5.18	Unknown environment	85	10.1

Table 5.7: Travelling path lengths comparison between simulation and experimental results

Figure no. (Simulation and experimental res.)	**Travelling path length (cm)**		**Error between simulation and experimental result**
	Simulation result	**Experimental result**	
Figures 5.6 and 5.16	103	109	5.5%
Figures 5.7 and 5.17	89	94	5.32%
Figures 5.13 and 5.18	80	85	5.88%

Table 5.8: Navigation time comparison between simulation and experimental results

Figure no. (Simulation and experimental res.)	**Navigation time (sec)**		**Error between simulation and experimental result**
	Simulation result	**Experimental result**	
Figures 5.6 and 5.16	11.6	12.4	6.45%
Figures 5.7 and 5.17	10.1	10.8	6.48%
Figures 5.13 and 5.18	9.4	10.1	6.93%

5.6 Summary

In this chapter, the CN-Fuzzy architecture has been applied to the intelligent navigation of a mobile robot in unknown environments filled with obstacles. The major contributions of this present chapter are summarized as follows: -

1) The cascade neural network is designed to train the robot to reach the goal in the environment. The inputs of cascade neural network are the obstacle distances, and the output is the turning angle between the robot and goal.
2) The fuzzy logic controller helps the robot to control the right motor velocity and left motor velocity in the environments for obstacle avoidance.
3) The proposed CN-Fuzzy architecture gives better results (in terms of path length) as compared to previous models [28] and [11], which proves the authenticity of the proposed architecture.
4) Moreover, the simulation and experimental results in the different environment show the effectiveness of the proposed architecture in the both static and dynamic environments. In the comparison study between the simulation and experiment results errors are recorded, and the errors are found due to the effect of slippage and friction between the wheels of the robot and surface during navigation in real time mode.
5) During experiment utmost care has been taken to minimize the slippage and friction between the wheels and surface. Still the effect of slippage and friction are unavoidable, and errors are recorded during the comparison of the results for travelling path length (5.57%) and for navigation time (6.62%).

Chapter 6

Mobile Robot Navigation in Different Environments using Takagi-Sugeno Fuzzy Controller and Simulated Annealing Algorithm Controller

6.1 Introduction

In this chapter, a Takagi-Sugeno fuzzy model with simulated annealing hybrid algorithm (Fuzzy-SA) has been designed and implemented for the mobile robot navigation and obstacle avoidance in the different environments. The simulated annealing algorithm is used to optimize the output value of the fuzzy controller. The ultrasonic range finder sensor and sharp infrared range sensor are used to calculate the different obstacle distances, such as front, right, and left obstacle distance for selecting the suitable steering angle control command in the environment. The objective function for the simulated annealing algorithm is considered based on the shortest path using the fuzzy model. The simulation and experimental results show the proposed method is feasible and valid for a wheeled mobile robot moving in the different environments.

Navigation can be defined as the process of directing the safe movement of a mobile robot from one point to another with the help of different types of sensors for obstacle detection in the different environments like indoor, outdoor, and cluttered by using the various artificial intelligence navigation techniques. The fuzzy logic plays an important role in dealing with uncertainty when decision making required for mobile robotic applications for intelligent control like steering control and motor control, etc. Path planning is a major topic in the field of a mobile robot, which means to figure out an optimal or near-optimal non-collision path from a start point to a goal point in an environment with the presence of different shape obstacles. In the present work, the simulated annealing algorithm has been integrated with the Takagi-Sugeno fuzzy controller to explore the collision-free path planning for a mobile robot. Moreover, the effectiveness and efficiency of the proposed hybrid method are demonstrated through MATLAB software simulation and also done in real-time experiments in various environments.

The application of mixed soft computing techniques such as Neural Network [11], Fuzzy Logic [28], Simulated Annealing Algorithm [108], Genetic Algorithm [165], Ant Colony Optimization Algorithm [144], Particle Swarm Optimization [18], Bee Algorithm [166], Cuckoo Search Algorithm [167] and other nature-inspired algorithms have been successfully applied by the various researchers in the field of mobile robotics. The concept of simulated annealing algorithm has come from statistical mechanics [104]. The simulated annealing is an iterative search algorithm inspired by the annealing of metals [105]. In the current study, the authors have attempted to solve the path optimization problem using a fuzzy controller with simulated annealing hybrid algorithm. The purpose of the navigation problem for the robot is to search an optimal path between the start point to goal point [95]. Simulated annealing is a stochastic optimization algorithm, and it is used for finding the global optimization value (either minimum or maximum) by the objective function [107]. Simulated annealing algorithm is suitable for the local minima problem avoidance [168].

In the present study, the application of the fuzzy model and simulated annealing mixed optimization algorithm has been applied to the mobile robot path planning problem. The contributions of this chapter are organized as follows: Section 6.1 presents the introduction. The design of Sugeno-type fuzzy logic model is presented in Section 6.2. Optimizing the fuzzy output using a simulated annealing algorithm for mobile robot navigation is proposed in Section 6.3. Section 6.4 and 6.5 presents the simulation results and discussion and comparison with previous works to the proposed hybrid controller, respectively. Section 6.6 represents the experimental setup and its results and discussion for validating the proposed hybrid controller. Finally, Section 6.7 depicts the summary.

6.2 Design of Sugeno-Type Fuzzy Logic Controller

In this section, the fuzzy controller is used to solve the navigation problem of a mobile robot in the presence of different shape obstacles. This fuzzy logic controller is activated when the robot detects any obstacle. The proposed membership functions and the fuzzy rule-based system are used to navigate the robot among the obstacles in any environment. Fuzzy logic is a method of formulating the mapping relationship between the inputs to an output using the mathematical concept. The Fuzzy Inference Systems (FIS) have been classified into two groups: Mamdani-type and Takagi-Sugeno type. The proposed fuzzy controller is used to control the steering angle of the mobile robot, which allows the robot to move from start point to goal point in the environment. The fuzzy controller has three input variables: x_1 *(Front Obstacle Distance),* x_2 *(Right Obstacle Distance),* and x_3 *(Left Obstacle Distance)* and single output: f *(Steering Angle)*. The proposed controller receives inputs (obstacle distances) from the various sensors of the robot. Two Generalized Bell-Shaped (gbell) membership functions (MFs) are considered for the inputs. The range of inputs is divided into two linguistic variables: near and away respectively. The two constant type membership functions (MFs) negative and positive respectively have been used for the output, and it is located at -90 and 90 respectively. Figures 6.2 and 6.3 show the membership parameters of the input and output variables respectively. The fuzzy controller is composed through zero-order Takagi-Sugeno model in the following form: -

$$Rule_i : If\ x_1\ is\ A_{i1},\ x_2\ is\ A_{i2},\ And\ x_3\ is\ A_{i3}\ THEN\ f_i\ is\ \alpha_i \tag{6.1}$$

where the symbols x_1, x_2, and x_3 are the input variables, A_{i1}, A_{i2}, and A_{i3} are the fuzzy sets, and α_i is a real number. The fuzzy set A_{ij} uses the following generalized bell-shaped membership function: -

$$\mu_{ij}\left(x_j\right) = \frac{1}{1+\left|\frac{x_j - c_{ij}}{a_{ij}}\right|^{2b_{ij}}} \tag{6.2}$$

where c_{ij} is the center of the generalized bell-shaped membership function, a_{ij} is the half width of the membership function, and b_{ij} (together with a_{ij}) is controlling the slopes at the crossover points. The general structure of the generalized bell-shaped membership function is shown in Figure 6.1. The firing strength $\delta_i(x)$ is calculated by the following function: -

$$\delta_i(x) = \prod_{j=1}^{n} \mu_{ij}(x_j) \tag{6.3}$$

The defuzzification of the output (steering angle) is accomplished by the weighted average method: -

$$f_i = \frac{\sum_{n=1}^{8} \delta_i(x) \cdot \alpha_i}{\sum_{n=1}^{8} \delta_i(x)} \tag{6.4}$$

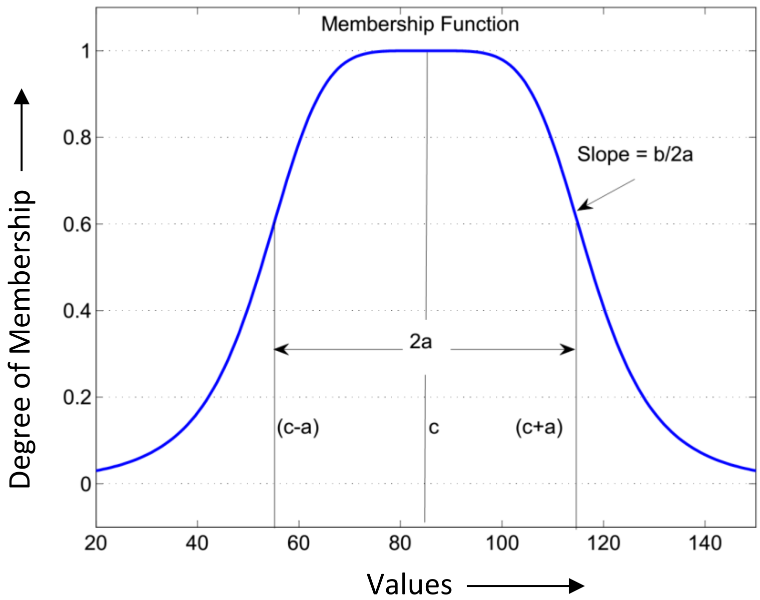


Figure 6.1: Generalized bell-shaped membership function.

To avoid the obstacles, the controller drives with user defined rule system based on human experience, and its functioning under Takagi-Sugeno type fuzzy controller is shown in Figure 6.4. Eight fuzzy control rules have been designed for the proposed navigation controller; and are listed in Table 6.1. From the table analysis (fuzzy rule number three) when the mobile robot comes near the left obstacle, then the robot will turn to the right side (negative steering angle) for reaching the goal without any collision. From the fuzzy rule number four, when the mobile robot comes near the right obstacle, then the robot will turn to the left side (positive steering angle) for reaching the goal without any collision. The negative steering angle control command means the right motor velocity will be low, and the left motor velocity will be high respectively. The positive steering angle control command means the right motor velocity will be high, and the left motor velocity will be low respectively.

The rule viewer of the proposed fuzzy controller is shown in Figure 6.5, which represents the eight fuzzy rules in the Graphical User Interface (GUI) form. The first three input column corresponds to the Front Obstacle Distance (F.O.D.), Right Obstacle Distance (R.O.D.), and Left Obstacle Distance (L.O.D.) respectively. These obstacle distances have received from the sensors. The fourth column represents the Steering Angle (S.A.), which is calculated by the weighted average method (equation (6.4)) of the fuzzy controller. The steering angle surface plot generated by the fuzzy controller is shown in Figure 6.6. The proposed fuzzy model has been generated through the MATLAB programming, which is described below (MATLAB Fuzzy Logic Function).

MATLAB FUZZY LOGIC FUNCTION

Program:

```
begin:
    fuzzy inference system
    fuzzification
    add variable for input1 (F.O.D.)
    add generalized bell-shaped (gbellmf) membership function for input1 (F.O.D.)
    add variable for input2 (R.O.D.)
    add generalized bell-shaped (gbellmf) membership function for input2 (R.O.D.)
    add variable for input3 (L.O.D.)
    add generalized bell-shaped (gbellmf) membership function for input3 (L.O.D.)
    add variable for output (S.A.)
    add Sugeno-type constant membership function for output (S.A.)
    rule base
    add rules
    defuzzification
    result crisp output (S.A.)
end
```

Table 6.1: Fuzzy control rules for mobile robot navigation using two-membership functions

Fuzzy rules	**Front Obstacle Distance (F.O.D.)**	**Right Obstacle Distance (R.O.D.)**	**Left Obstacle Distance (L.O.D.)**	**Steering Angle (S.A.)**
1	Away	Away	Away	Negative
2	Near	Near	Near	Positive
3	Away	Away	Near	Negative
4	Away	Near	Away	Positive
5	Near	Away	Away	Positive
6	Near	Away	Near	Negative
7	Near	Near	Away	Positive
8	Away	Near	Near	Positive

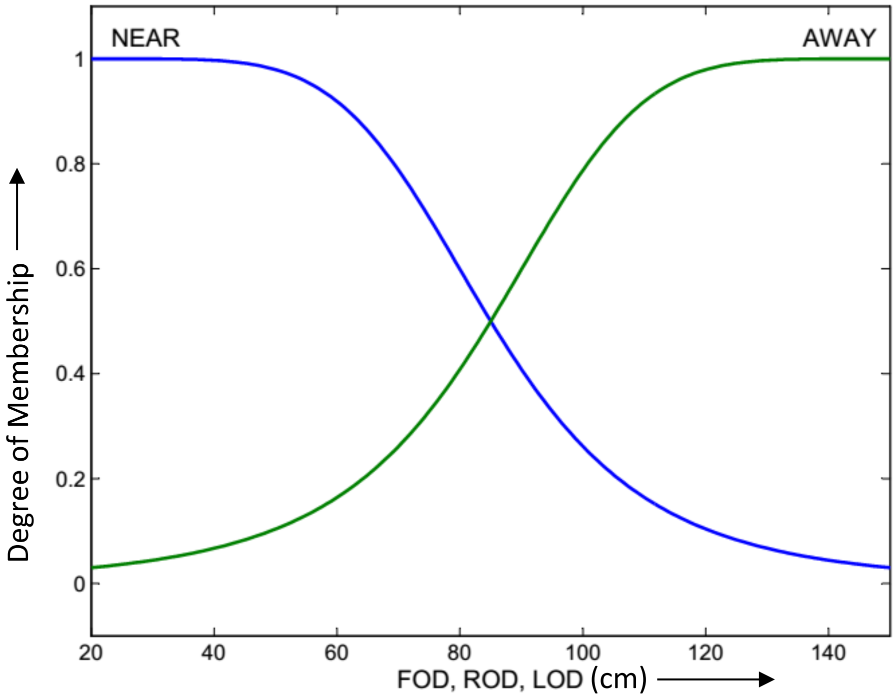


Figure 6.2: Fuzzy membership functions for the inputs (F.O.D., R.O.D., and L.O.D.).

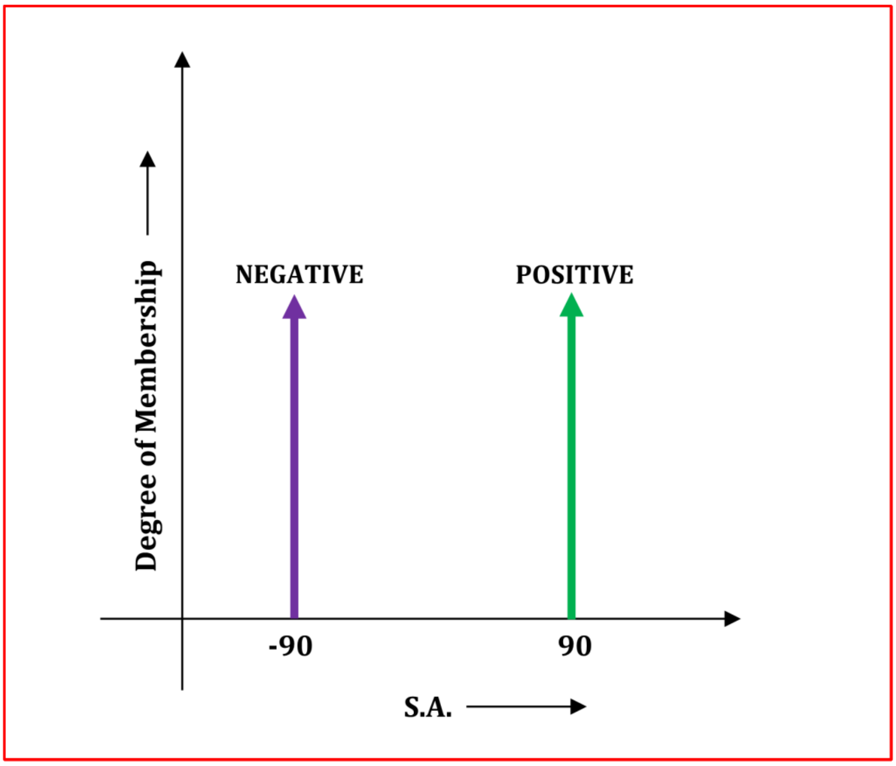


Figure 6.3: Fuzzy membership function Sugeno-Type for output variable steering angle.

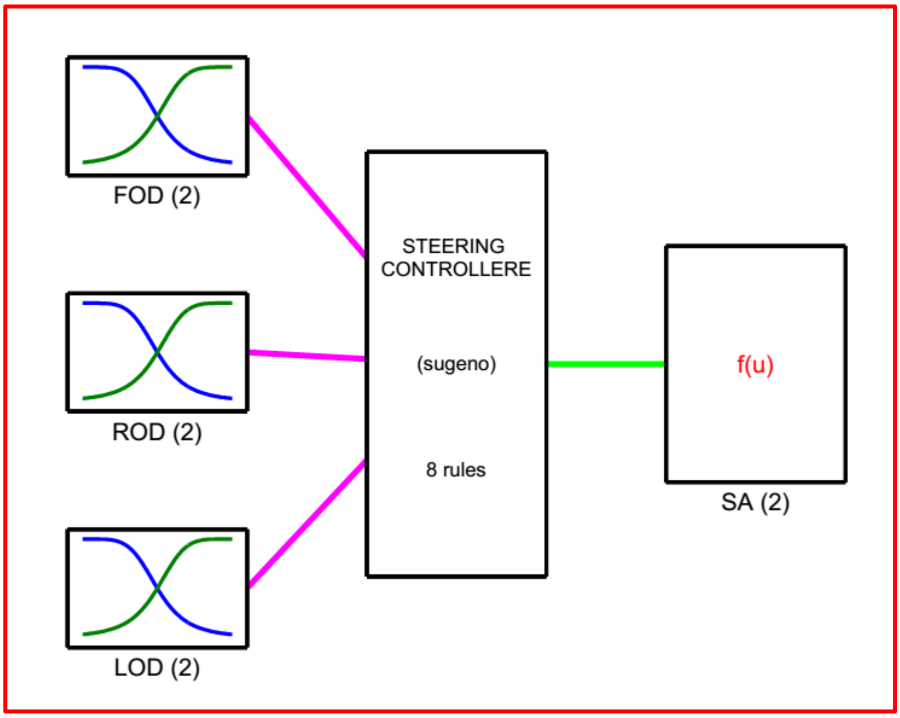


Figure 6.4: Takagi-Sugeno type fuzzy controller.

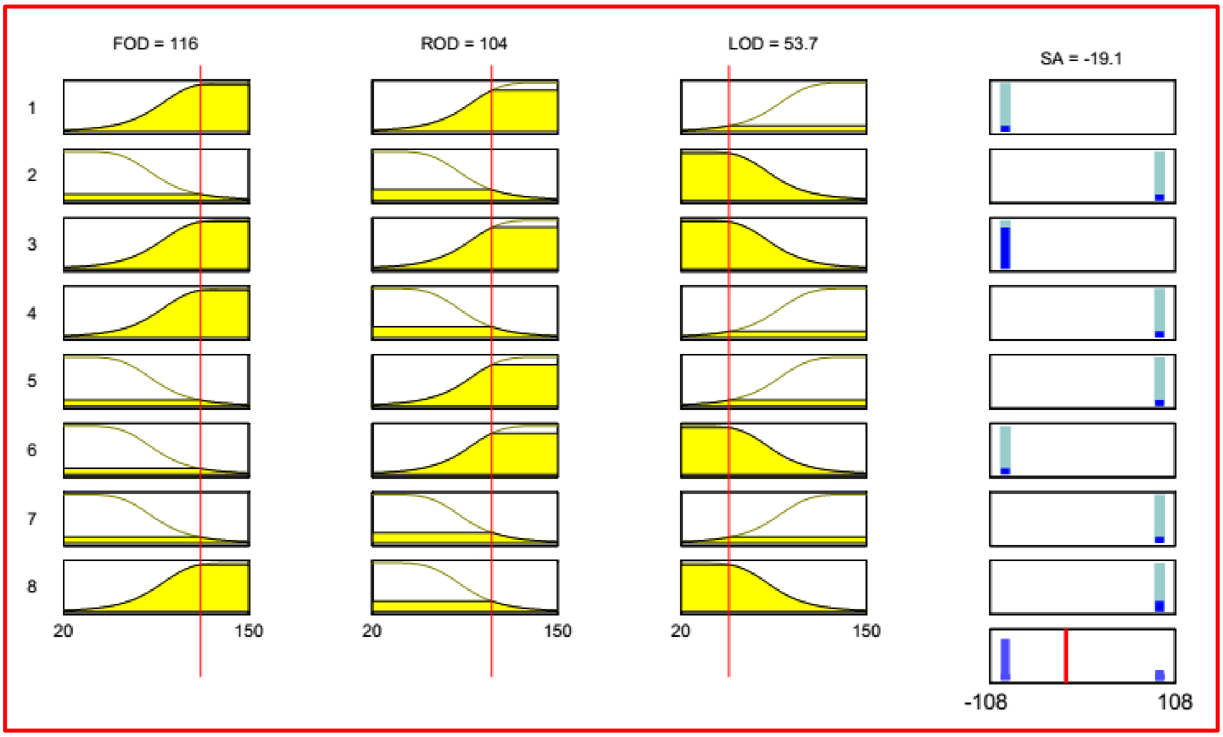


Figure 6.5: Rule viewer of the fuzzy controller.

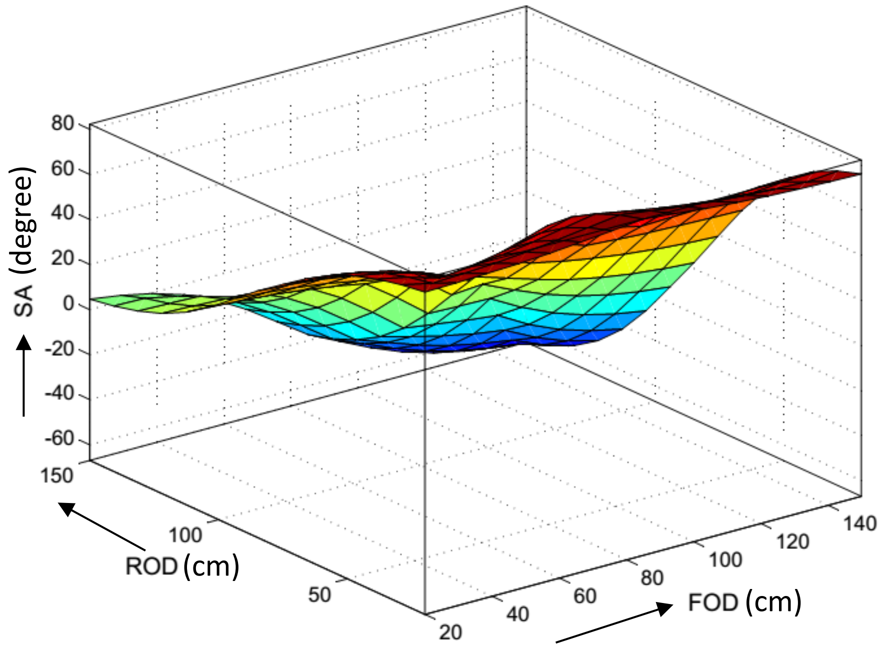


Figure 6.6: Steering angle control surface function plot.

6.3 Optimizing the Fuzzy Controller Output using Simulated Annealing Algorithm (SAA)

Simulated Annealing Algorithm (SAA) is an iterative search metaheuristic algorithm, which is inspired by the annealing of metals. SAA is a stochastic optimization algorithm, and is used for finding the global optimization value (either minimum or maximum) by the objective function. The SAA starts from an initial solution at a high temperature and makes some changes according to annealing schedules. For any two iterations, there are two objective values marked as f_i (new) and f_{i-1} (old), and the difference between the objective values $(\Delta f = f_i - f_{i-1})$ is calculated. If $\Delta f \leq 0$, then the new solution is accepted with probability $p(T) = 1$. Otherwise, it is accepted with small probability $p(T)$, $p(T) = \exp\left(\frac{-\Delta f}{k \cdot T}\right)$, where k is the constant (Boltzmann) parameter of the process and T is the instantaneous temperature. After enough numbers of iteration, we get the optimum value of the solution. During SAA process, the objective is move from the high energy region to low energy region in the search space.

In this section, the result of fuzzy output (consequent) has been optimized through the simulated annealing algorithm. The fuzzy controller is used to receive the obstacle distance from the group of sensors to control the steering angle of the wheeled mobile robot. And the simulated annealing algorithm is applied to find the optimum steering angle of the mobile robot by using an objective function. The parameters C_1, C_2, C_3, and C_4 are optimized through SAA to acquire optimum steering angle for mobile robot navigation using objective function. This optimum steering angle is helping to achieve the minimum path length in the environment. The following objective function calculates the optimal steering angle (O.S.A.) for a mobile robot: -

$$O.S.A. = C_1 * F.O.D. + C_2 * R.O.D. + C_3 * L.O.D. + C_4 * S.A. \tag{6.5}$$

where,

$20 \leq F.O.D. \leq 150$

$$20 \le R.O.D. \le 150$$

$$20 \le L.O.D. \le 150$$

$$-90 \le S.A. \le 90$$

Here the fuzzy rule-based controller has been integrated with the SAA controller to optimize the steering angle of the mobile robot in different environments. The resulting Fuzzy-SA controller for navigation of the mobile robot is shown in Figure 6.7. The role of the fuzzy controller is to estimate the initial steering angle for the hybrid controller. Then, this initial steering angle is fed to an SAA controller along with the obstacle distance (front, right and left), and finally this hybrid controller does optimize the steering angle using a fitness function (equation (6.5)). Figure 6.8 shows the objective function value versus the number of iteration number. The simulated annealing algorithm model has been made by MATLAB function programming, which is listed below (MATLAB Simulated Annealing Algorithm (SAA) Function).

MATLAB SIMULATED ANNEALING ALGORITHM (SAA) FUNCTION

Program:

function:

Y = new_simulated_annealing(X, F.O.D., R.O.D., L.O.D., S.A.)

$C_1 \leftarrow X(1);$

$C_2 \leftarrow X(2);$

$C_3 \leftarrow X(3);$

$C_4 \leftarrow X(4);$

*Y = X (1)*F.O.D. + X (2)*R.O.D. + X (3)*L.O.D. + X (4)*S.A.;*

end function

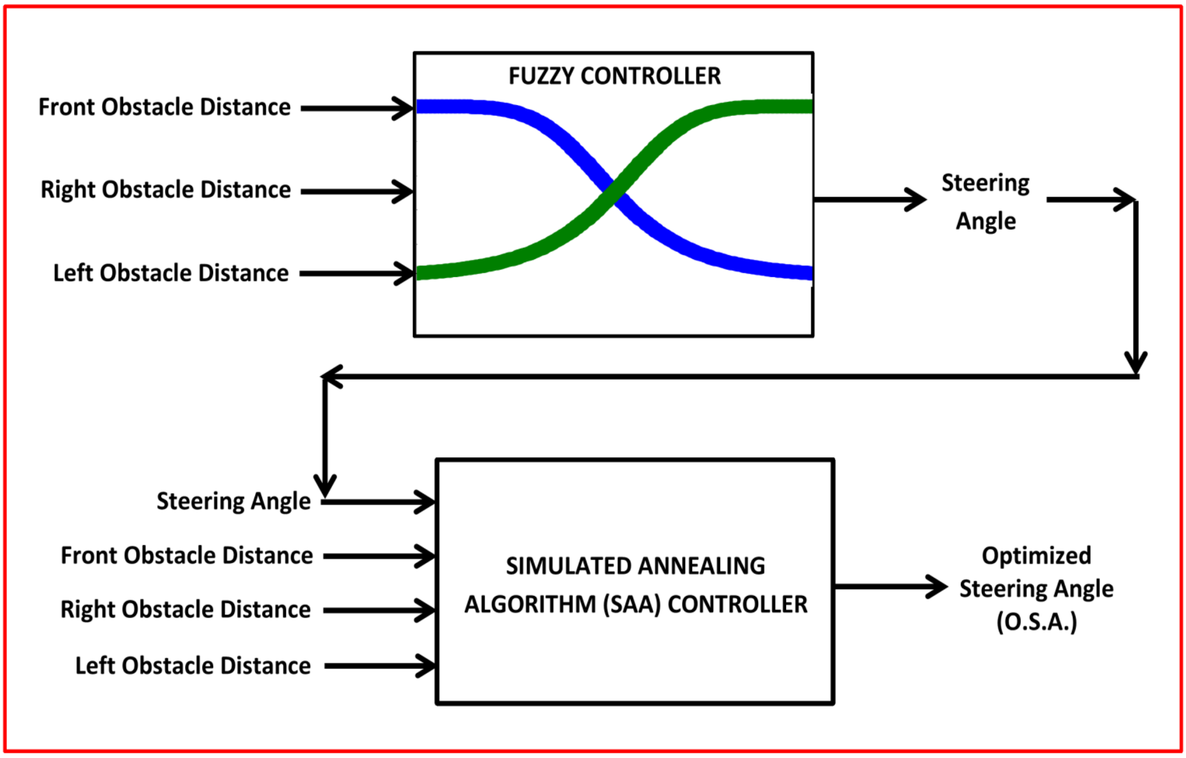


Figure 6.7: Fuzzy-SA controller for navigation of a mobile robot.

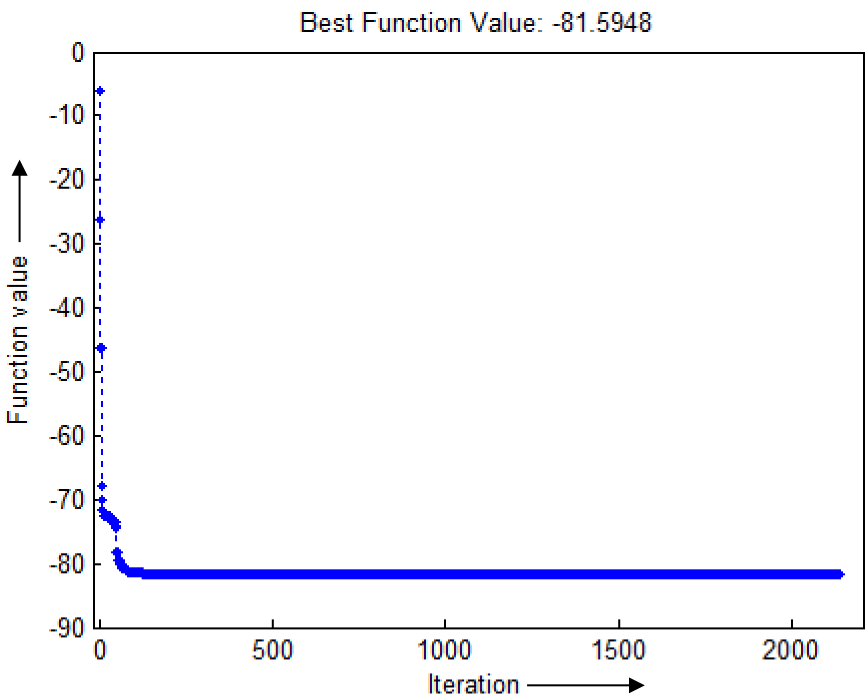


Figure 6.8: Objective function value versus number of iteration number.

6.4 Simulation Results and Discussion

In this section, simulations are presented to demonstrate the effectiveness of the singleton fuzzy controller and Fuzzy-SA controller. The simulation results are performed by using MATLAB software. The developed architecture of mobile robot navigation based on Fuzzy-SA algorithm is given in Figure 6.9. Figures 6.10 to 6.12 show the mobile robot navigation and obstacle avoidance in the different environments using singleton fuzzy controller and Fuzzy-SA controller. Similarly, the Figure 6.13 demonstrates the navigation of a mobile robot in an unknown environment with the presence of two dynamic obstacles using Fuzzy-SA controller. It is assumed that the position of the start point and goal point are known. But the positions of all the obstacles in the environment are unknown for the robot. The red line indicates the mobile robot navigation using singleton fuzzy controller and the green line indicates the mobile robot navigation using Fuzzy-SA controller. The following criteria are used to evaluate the performance comparison between the singleton fuzzy controller and the Fuzzy-SA controller: -

(1) Navigation path length (from the start point to goal point).
(2) Navigation time (from the start point to goal point).

The comparison of results between the singleton fuzzy controller and the Fuzzy-SA controller has been listed in Table 6.2. From the table, it can be seen that the proposed Fuzzy-SA controller gives better results compared to the singleton fuzzy controller.

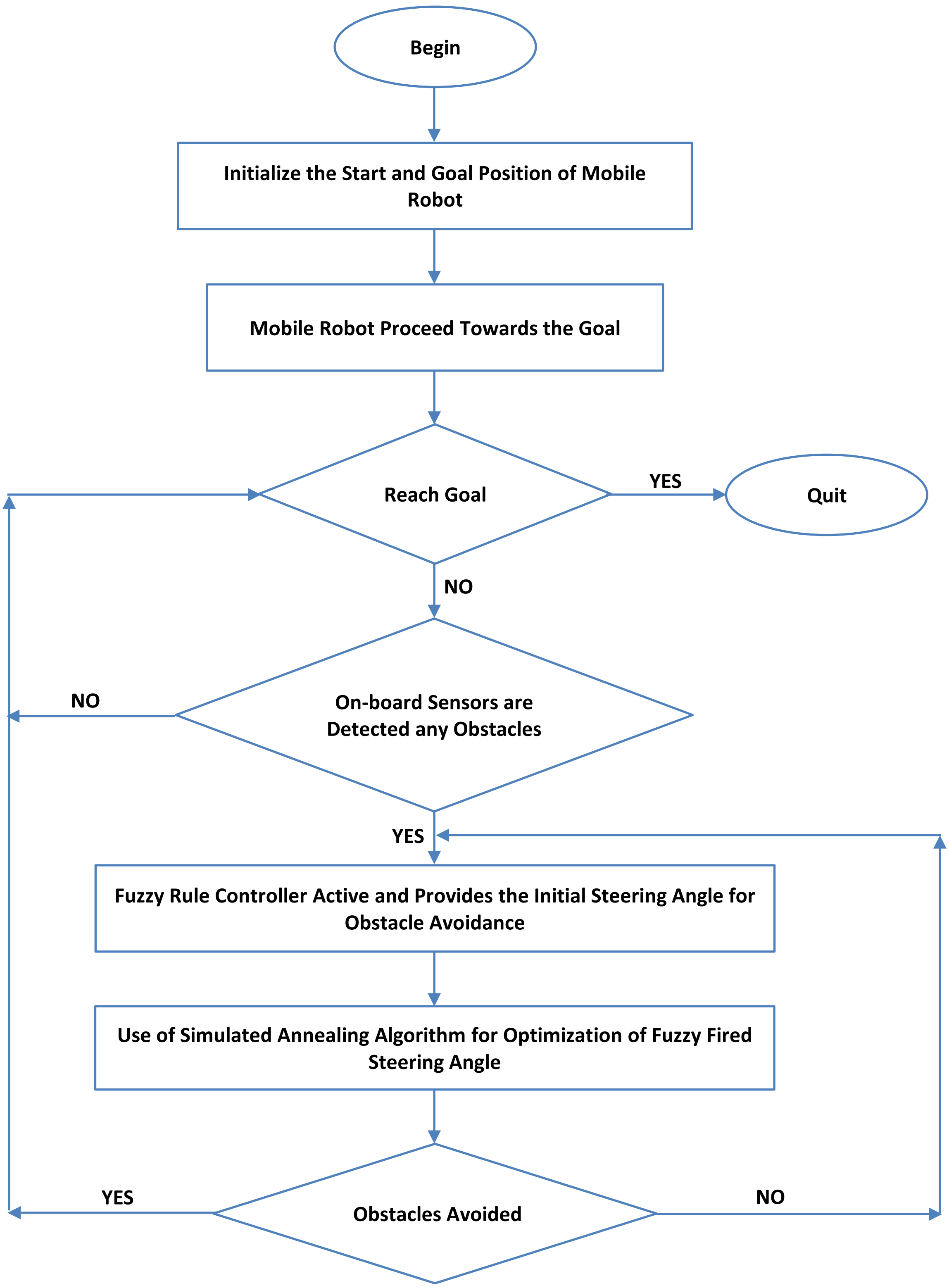
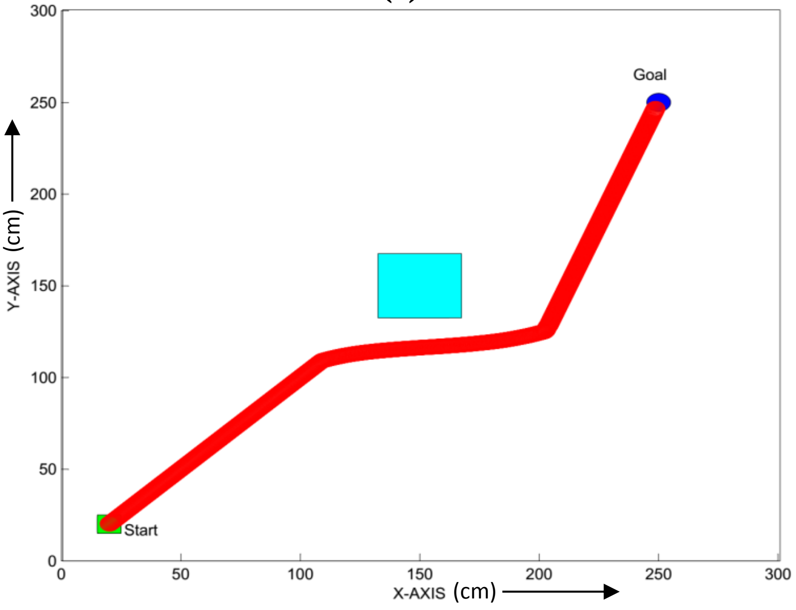


Figure 6.9: The developed architecture of mobile robot navigation based on Fuzzy-SA algorithm.

(a)



(b)

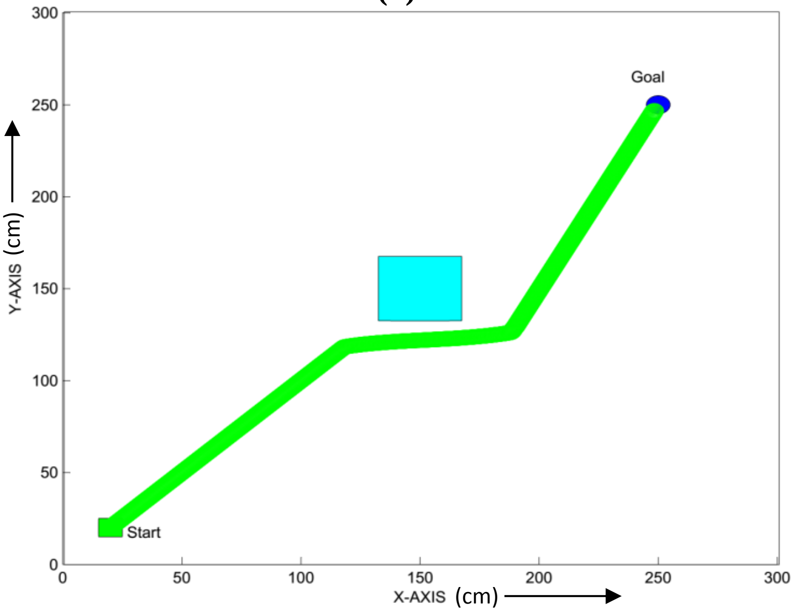


Figure 6.10: Mobile robot navigation among the single obstacle using **(a)** Fuzzy controller and **(b)** Fuzzy-SA controller.

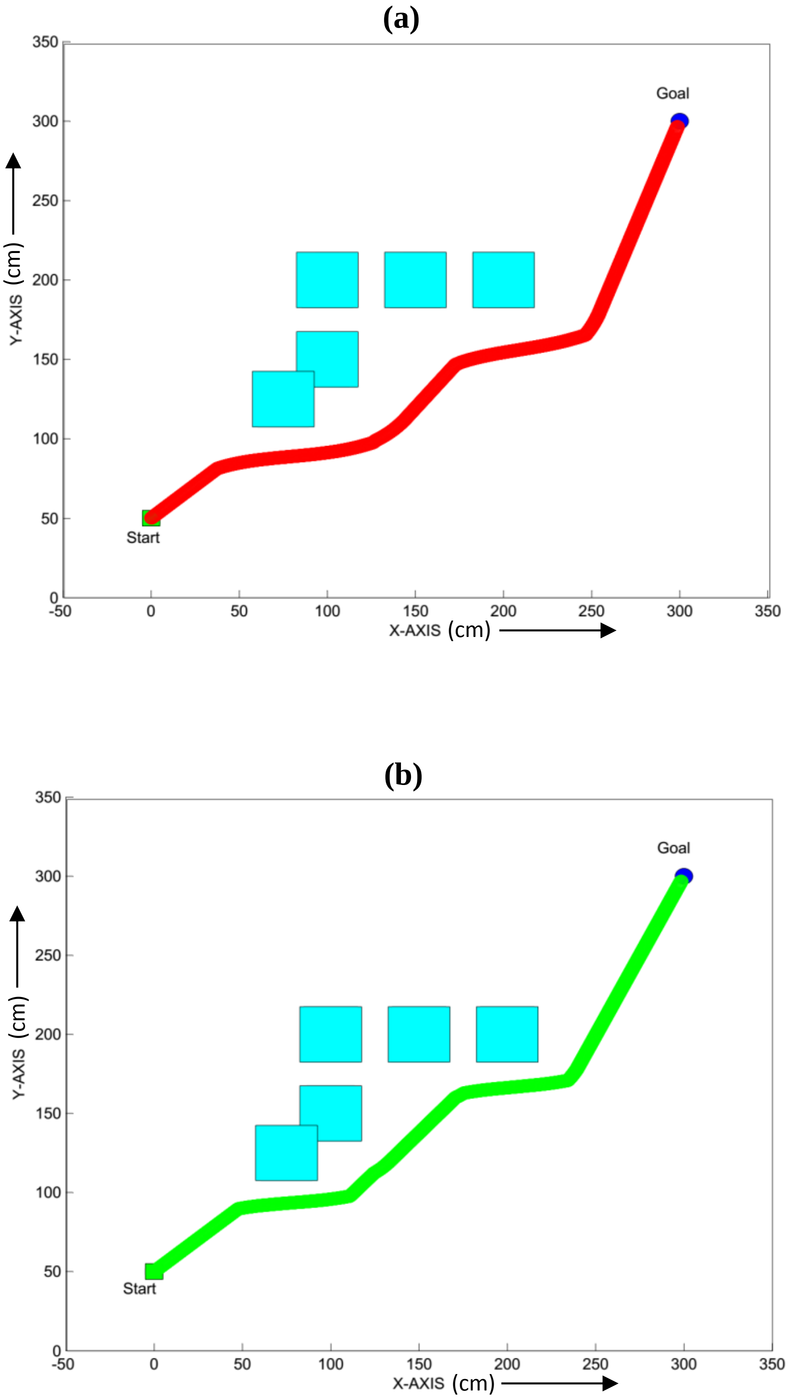
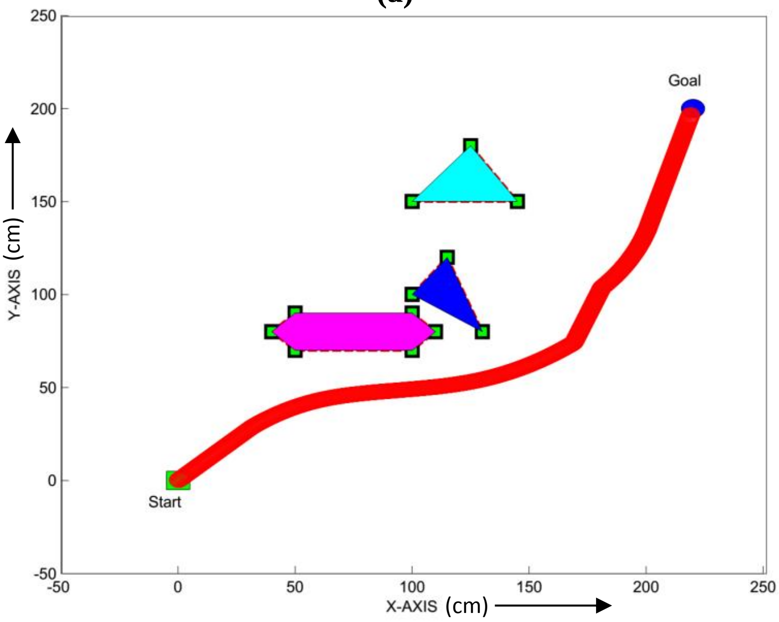


Figure 6.11: Mobile robot navigation among the many obstacles using **(a)** Fuzzy controller and **(b)** Fuzzy-SA controller.

(a)



(b)

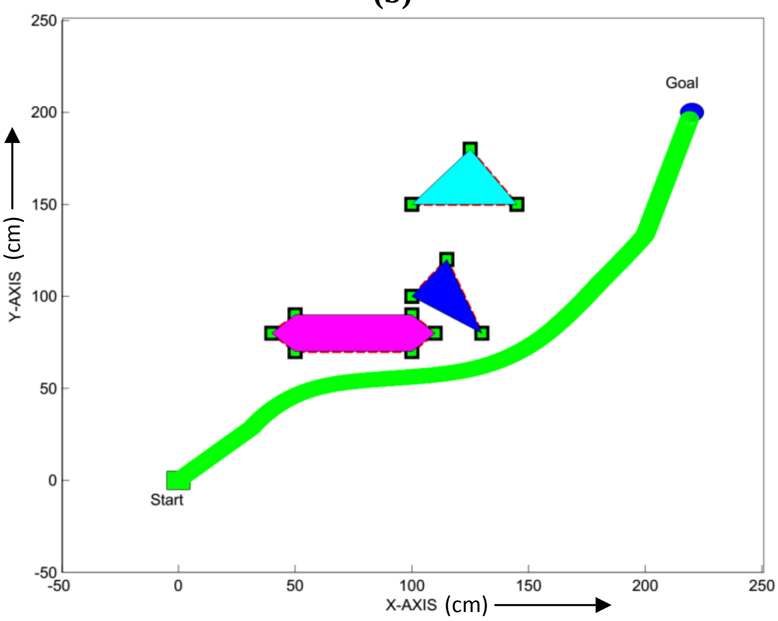


Figure 6.12: Mobile robot navigation among the polygonal obstacles using **(a)** Fuzzy controller and **(b)** Fuzzy-SA controller.

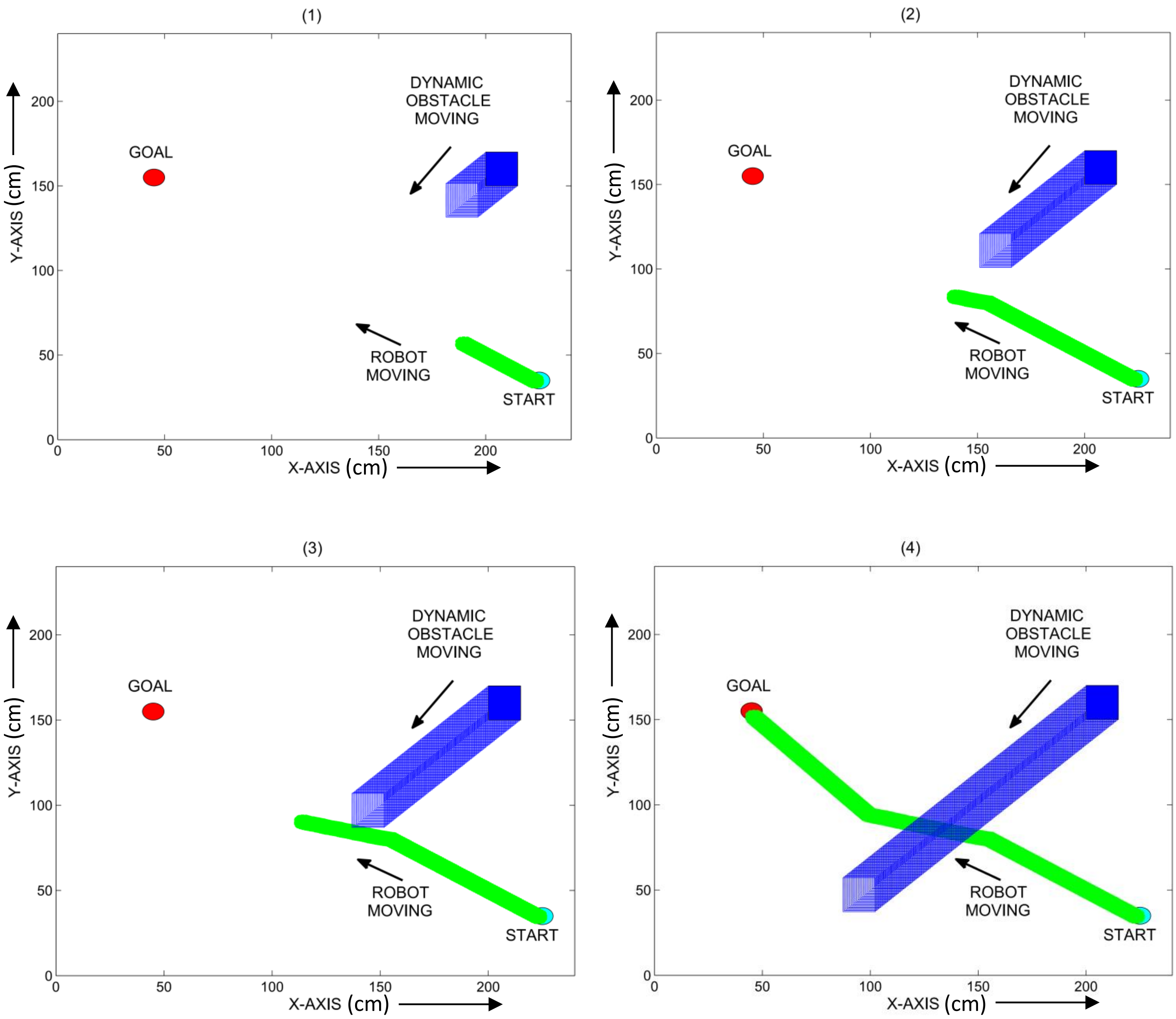


Figure 6.13: Mobile robot navigation in the dynamic environment using Fuzzy-SA controller.

Table 6.2: The result comparison between the singleton fuzzy controller and the Fuzzy-SA controller

Figure no.	Controller	Navigation path length (cm)	Percentage deviation of navigation path length	Navigation time (sec)	Percentage deviation of navigation Time
Figure 6.10 **(a)**, **(b)**	Fuzzy	126	1.61	8.4	2.44
	Fuzzy-SA	124		8.2	
Figure 6.11 **(a)**, **(b)**	Fuzzy	135	2.27	9.2	4.54
	Fuzzy-SA	132		8.8	
Figure 6.12 **(a)**, **(b)**	Fuzzy	118	3.51	7.9	5.33
	Fuzzy-SA	114		7.5	

6.5 Comparison with Previous Works

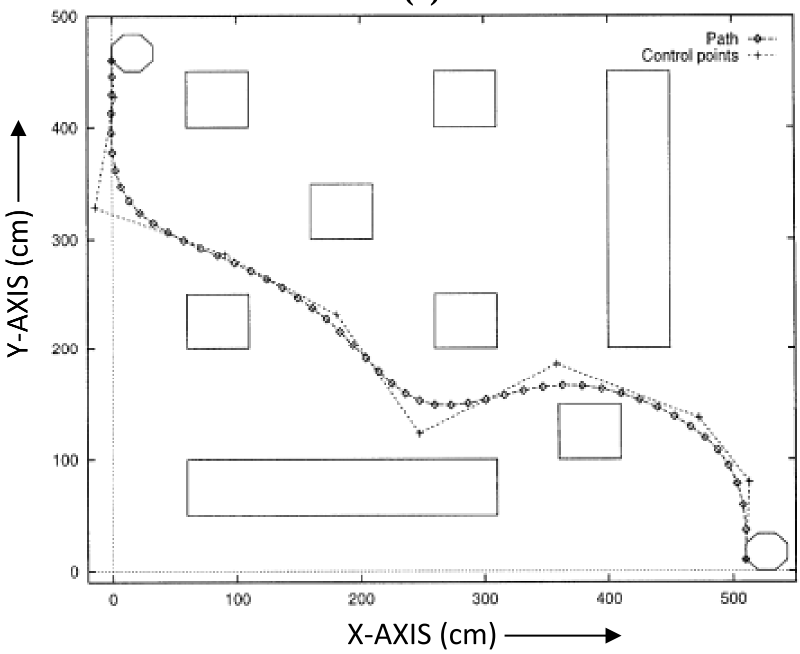
In this section, a comparison has been made between the proposed Fuzzy-SA controller and the previous models [107], [37] in the simulation and experimental modes. The performance of this hybrid model is evaluated on the basis of navigation path length.

Martinez-Alfaro et al. [107] have developed the simulated annealing and fuzzy logic for generating an automatic path planning of the mobile robot. The simulated annealing algorithm is used to search a collision-free optimal trajectory between the fixed polygonal obstacles. Forty-nine fuzzy rules have prepared to adjust the velocity of the robot based on the sonar readings of the obstacle. Liu et al. [37] have studied the path planning problem of an autonomous mobile robot based on ultrasonic range finder sensor information by combining the genetic algorithm with the fuzzy inference system.

Figures 6.14 **(a)** and 6.14 **(b)** present the path covered by the robot using the previous model [107] and the proposed hybrid model respectively, in the same environment. Similarly, Figures 6.15 **(a)** and 6.15 **(b)** illustrate the simulation result of the fuzzy-genetic model [37] and the proposed hybrid model respectively, in the same environment.

Martinez-Alfaro et al. [107] model (Table 6.3) robot finds the goal with the navigation path length of 266cm, and the proposed hybrid model finds the goal with the optimized navigation path length of 252cm. In Liu et al. [37] method (Table 6.4) robot finds the goal with the navigation path length of 44cm, and the proposed hybrid model finds the goal with the optimized navigation path length of 39cm. From the table analysis below, it can be clearly seen that the proposed Fuzzy-SA controller gives better results compared to the previous models. The centimeter measurements are taken on the proportional basis.

(a)



(b)

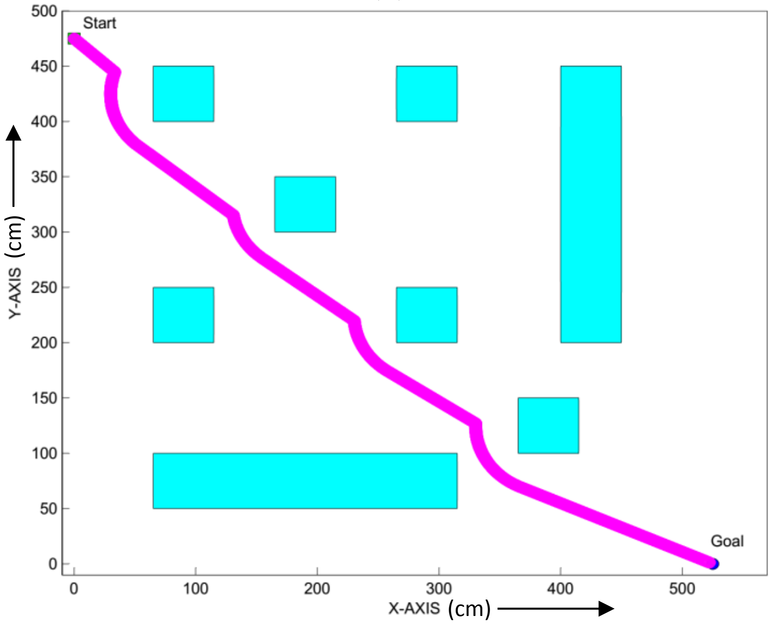
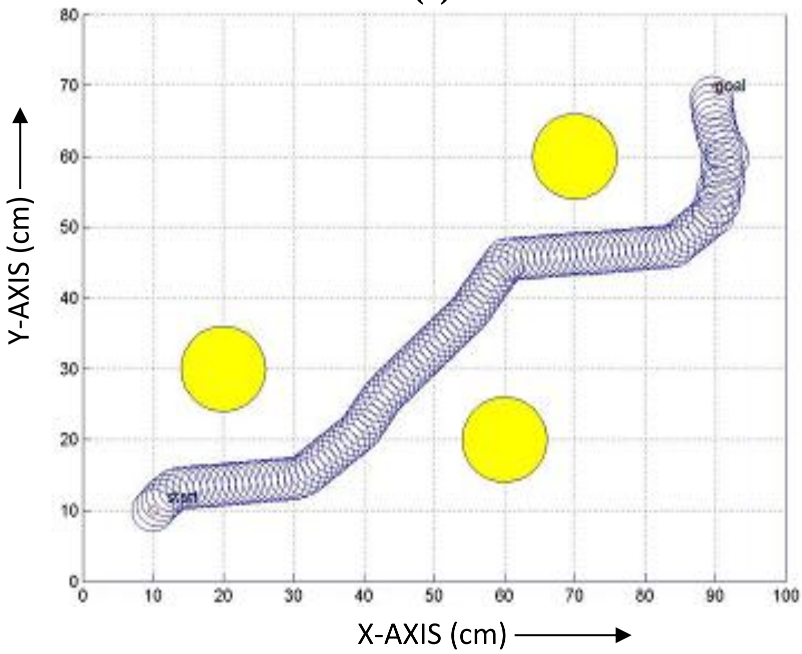


Figure 6.14: The graphical comparison between the **(a)** Martinez-Alfaro et al. [107] model and **(b)** Proposed hybrid model.

Table 6.3: The simulation result comparison between the Martinez-Alfaro et al. [107] model and proposed hybrid model

Figure no.	**Navigation path length of Martinez-Alfaro et al. [107] model (cm)**	**Optimized navigation path length in (cm)**
Figure 6.14 **(a)**, **(b)**	266	252

(a)



(b)

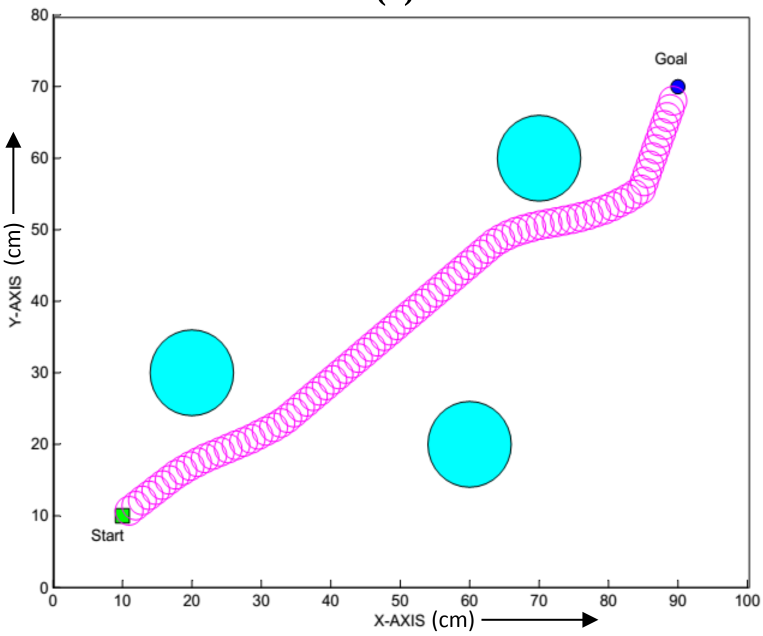


Figure 6.15: The graphical comparison between the **(a)** Liu et al. [37] model and **(b)** Proposed hybrid model.

Table 6.4: The result comparison between the Liu et al. [37] model and proposed hybrid model

Figure no.	**Navigation path length of Liu et al. [37] model (cm)**	**Optimized navigation path length (cm)**
Figure 6.15 **(a)**, **(b)**	44	39

6.6 Experimental Results and Discussion

6.6.1 Mobile Robot Description

To demonstrate the effectiveness of the proposed model, real-time experiments have been conducted using a two-wheeled mobile robot (Figure 6.16) in unknown environments. The robot has two front wheels driven by two center shaft 12Volt DC geared motors separately used for steering angle or driving (motor speed) control, and one omnidirectional caster wheel has been attached on the back side for balance. Two separate 12Volt DC motors are connected to a dual DC motor controller, and the microcontroller is used to drive each front wheel to facilitate turn left and right, backward and forward movements, and the motor voltage regulated by Pulse Width Modulation (PWM) signal. The width of the robot plate is 23cm, the distance between the wheels is around 30cm, wheel diameter is 10cm, and width of the wheels is 4cm. The mobile robot is equipped with one sharp infrared range sensor on the front side, and the two ultrasonic range finder sensors are equipped on the left and right side of the robot, as shown in Figure 6.17. Each sensor reads ranges from 20 cm to 4m approximately. The maximum, and minimum velocity of mobile robot used for navigation is V_{max} = 0.167 m/sec, and V_{min} = 0.067 m/sec respectively.

6.6.2 Experiments

In during the experiments, the mobile robot is controlled by intelligent processor. The width and height of the experimental platform are 350cm and 350cm, respectively. The environment is assumed to be entirely unknown for the robot, and the sensing range of the on-board robot sensors is limited. The average moving speed of the robot is 0.09 m/sec. The experimental verification of the above simulation results has been shown in Figure 6.18 and Figure 6.19. In Figure 6.18, the single obstacle is used for the navigation. Similarly, in Figure 6.19, three obstacles are used for the navigation. It is assumed that the position of the start point and goal point are known, but the positions of all the obstacles in the environment are unknown for the robot. From the experimental analysis, it can be clearly seen that the proposed Fuzzy-SA controller successfully avoid the obstacles in the given environment. The navigation path length result between the simulation and experimental is listed in Table 6.5. Tables 6.6 and 6.7 illustrate the travelling path length and navigation time comparison between the simulation and experimental results. In the comparison study between the simulation and experiments, it is observed that some errors have been found, these are happened due to slippage and friction during real time experiment.

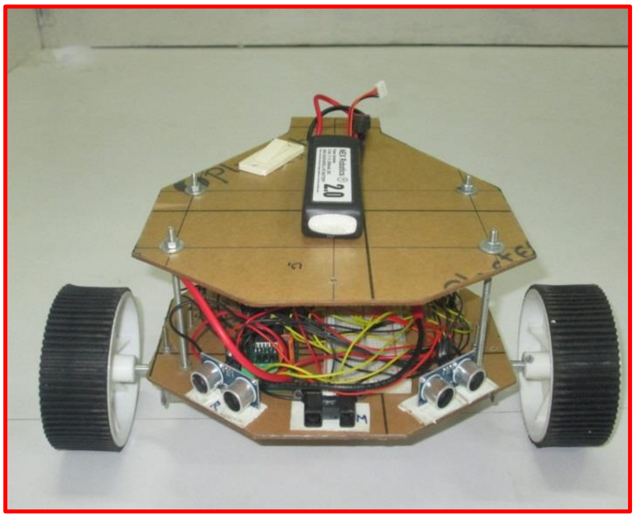


Figure 6.16: Two-wheeled mobile robot.

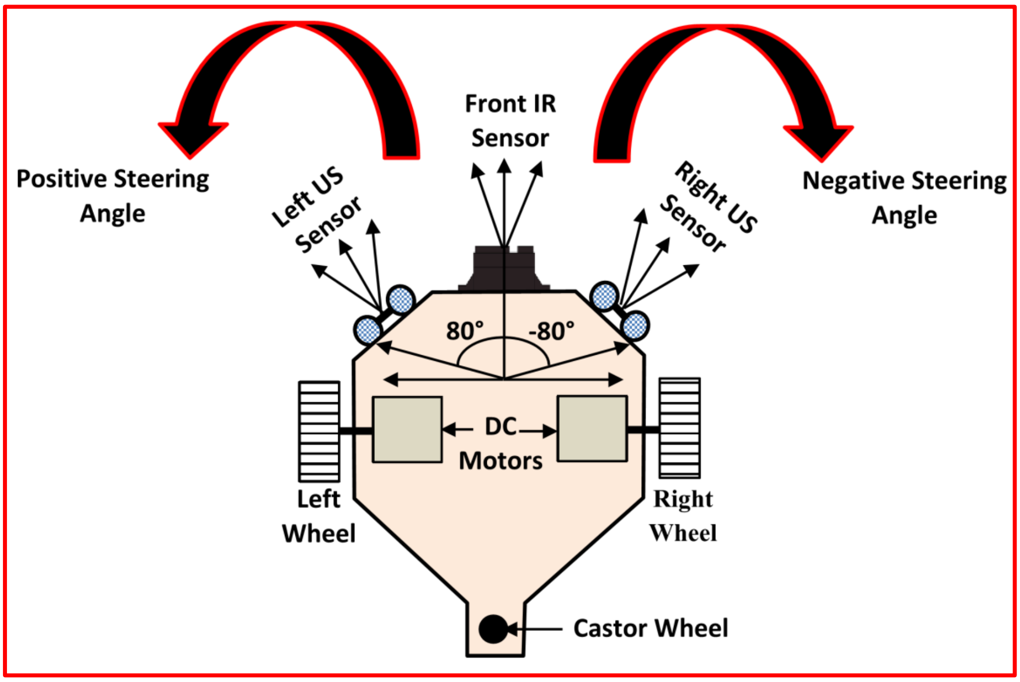


Figure 6.17: The arrangement of the sensors of a mobile robot.

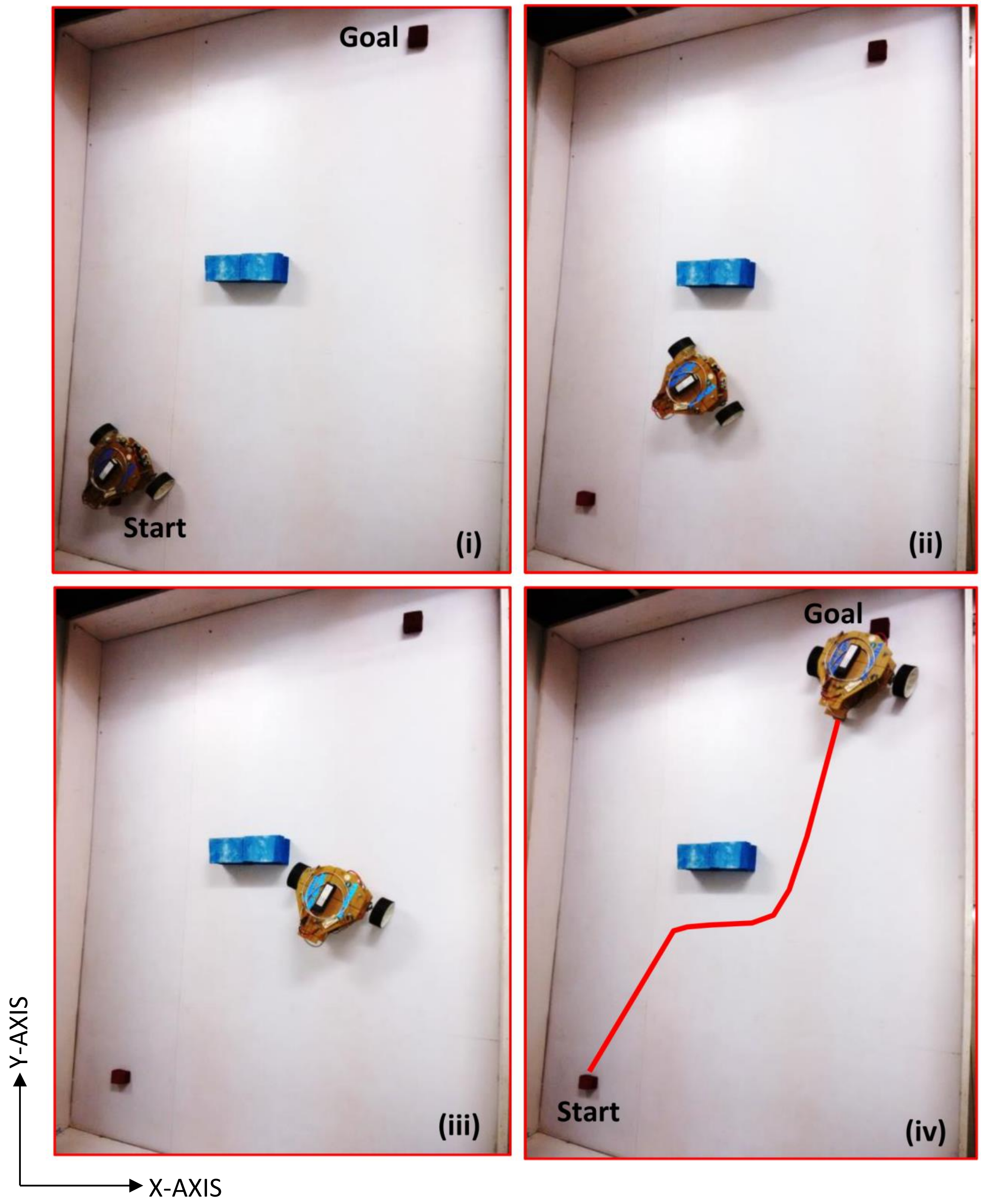


Figure 6.18: Experimental result of mobile robot navigation same as a simulation result (shown in Figure 6.10 (b)).

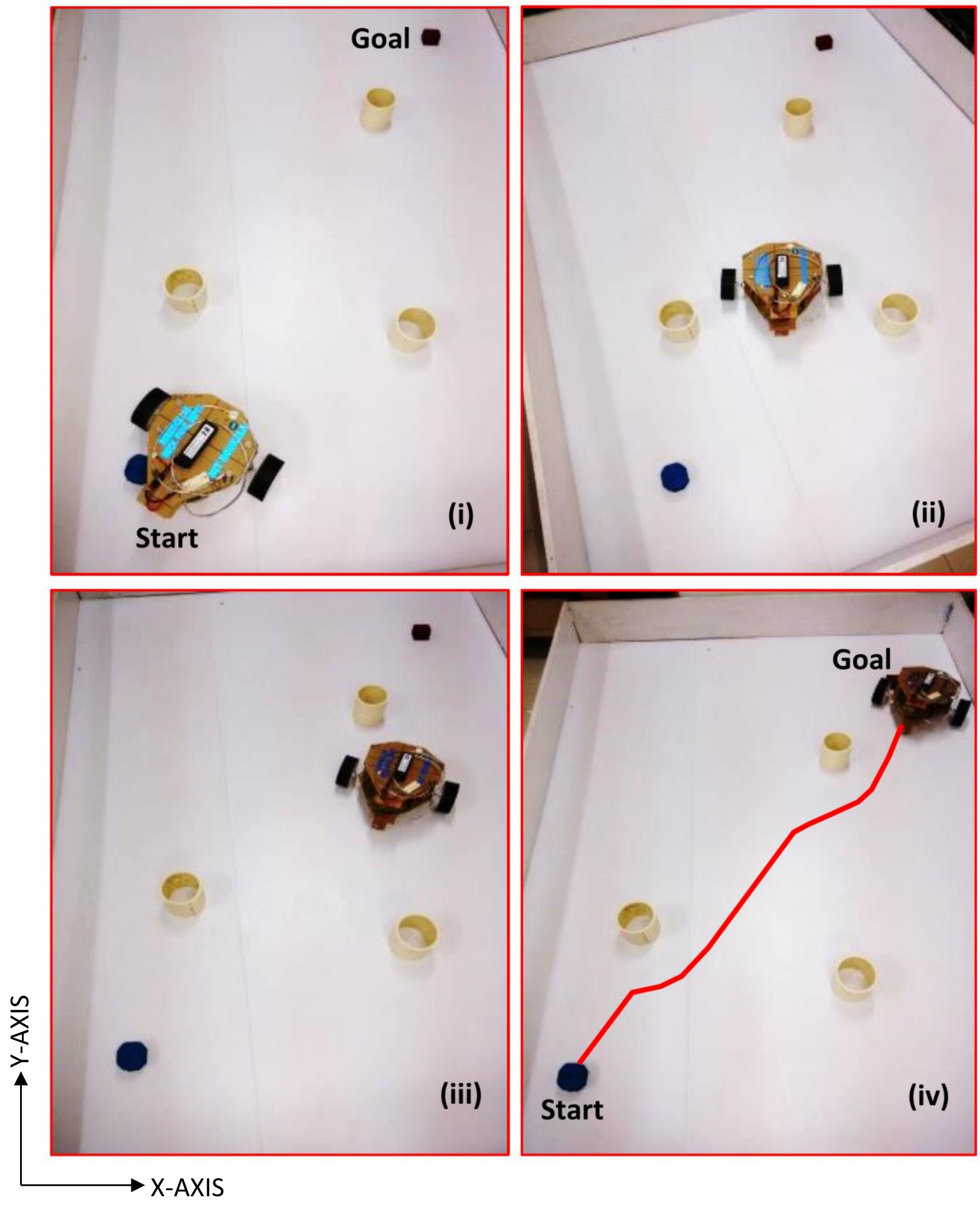


Figure 6.19: Experimental result of mobile robot navigation same as a simulation result (shown in Figure 6.15 (b)).

Table 6.5: Navigation path lengths between simulation and experimental results

Figure no. (Simulation and experimental res.)	Navigation Path Length (cm)	
	Simulation Result	Experimental Result
Figure 6.10 (b) and 6.18	124	131
Figure 6.15 (b) and 6.19	39	42

Table 6.6: Travelling path lengths comparison between simulation and experimental results

Figure no. (Simulation and experimental res.)	Travelling path length (cm)		Error between simulation and experimental result
	Simulation result	Experimental result	
Figures 6.10 (b) and 6.18	124	131	5.34%
Figures 6.15 (b) and 6.19	39	42	7.14%

Table 6.7: Navigation time comparison between simulation and experimental results

Figure no. (Simulation and experimental res.)	Navigation time (sec)		Error between simulation and experimental result
	Simulation result	Experimental result	
Figures 6.10 (b) and 6.18	8.2	8.8	6.82%
Figures 6.15 (b) and 6.19	4.6	4.9	6.12%

6.7 Summary

In the current study, a Takagi-Sugeno fuzzy model and simulated annealing hybrid algorithm (Fuzzy-SA) have been proposed for the mobile robot navigation and obstacle avoidance in the different environments (static and dynamic). The role of the fuzzy controller is to estimate the initial steering angle of the mobile robot, and the simulated annealing algorithm is applied to optimize this initial steering angle using objective function (see the equation (6.5)). Effectiveness and efficiency of this proposed Fuzzy-SA controller have been verified through many simulation and experimental studies in the real environment. In the comparison study between the simulation and experiment results errors are recorded, and the errors are found due to the effect of slippage and friction between the wheels of the robot and surface during navigation in real time mode. During experiment utmost care has been taken to minimize the slippage and friction between the wheels and surface. Still the effect of slippage and friction are unavoidable, and errors are recorded during the comparison of the results for travelling path length (6.24%) and for navigation time (6.47%). Moreover, this Fuzzy-SA controller is compared with previous Fuzzy-Simulated Annealing Algorithm [107] and the Fuzzy-Genetic Algorithm [37] to prove the authenticity of the proposed controller.

Chapter 7

Optimum Navigation of a Mobile Robot in the Different Environments using Wind Driven Optimization Algorithm

7.1 Introduction

In this chapter, atmospheric motion based Wind Driven Optimization (WDO) algorithm is implemented to optimize the navigation path length of a mobile robot in the various environments with the presence of different shape obstacles. This optimization algorithm is working based on the atmospheric motion of infinitesimal small air parcels navigates over an N-dimensional search domain. The WDO will be used to find the optimum or near-optimum steering angle for the navigation of a mobile robot to achieve the minimum path length in given environment. The consideration of the objective function is depended mainly for two reasons, i.e. obstacle avoidance and to find out the shortest possible path in an environment. Simulation and experimental results demonstrate that the robot can determine the optimum path length using Wind Driven Optimization (WDO) compared to the path length obtained by the robot using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) algorithm.

Metaheuristic optimization methods such as Genetic Algorithm [169], Simulated Annealing Algorithm [107], Particle Swarm Optimization [121], Ant Colony Optimization [148], Cuckoo Search [167], Firefly Algorithm [170], and many other nature-inspired optimization algorithms have been proposed and successfully implemented for the robot path planning problems in static as well as dynamic environments, but each algorithm possesses strengths and weakness. In the recent years, there are many researchers focused on meta-heuristic optimization in the field of mobile robot path optimization problems. Other soft computing techniques like Fuzzy logic

[171], Neural Network [80], ANFIS [172], and its hybrid methods [107,15] are also applied successfully in the various research area of mobile robot path planning in different environments. In this chapter, new type atmospheric motion based, a nature-inspired global optimization technique is implemented for mobile robot navigation in the unknown static and dynamic environments. The Wind Driven Optimization [153-154] method is a population-based iterative heuristic optimization algorithm for multi-dimensional and multi-model problems with the potential to implement constraints on the search domain [156]. As a newly intelligent optimization algorithm, WDO algorithm has been successfully applied in the different optimization fields [155-157] and achieved better results in some cases.

Path planning for a mobile robot finds an optimum and safe path from start point to a target point in an environment with obstacles. In this chapter, a new nature-inspired atmospheric motion based WDO algorithm is applied to solve the path planning problem of the mobile robot in unknown cluttered environments. WDO algorithm works by simultaneously maintaining several infinitesimal small air parcels or potential solutions in the search domain. For each iteration of the algorithm, each air parcels are evaluated by the objective function being optimized based on the fitness function of that solution. Consideration of the objective function is based on two reasons, i.e. obstacle avoidance and find out the shortest possible path in an environment. In the present study, an attempt is made to verify the effectiveness of the WDO algorithm based optimized navigation of a robot in unknown cluttered and dynamic environments filled with different shape obstacles. This optimization algorithm has been successfully implemented in the different areas of the engineering applications [155-157]. Due to its broad area of application and performance, this optimization algorithm has been used for solving the mobile robot's optimum path planning problem. In order to demonstrate the success of this new optimization technique, it is applied to different unknown simulation and experimental conditions and compared with previous techniques like adaptive Genetic Algorithm [169], Particle Swarm Optimization [173] and found to be good agreement in terms of navigation path length.

This chapter is structured as follows: Section 7.1 presents the introduction. Path optimization using Wind Driven Optimization (WDO) algorithm for mobile robot navigation is proposed in Section 7.2. Section 7.3 presents the computer simulation results and discussion. Comparison with the previous navigational controller with the proposed controller is listed in Section 7.4. Section 7.5 represents the experimental setup and its results. Finally, Section 7.6 depicts the summary.

7.2 Path optimization using Wind Driven Optimization (WDO) algorithm

Wind driven optimization [153] algorithm is inspired from the earth's atmosphere, where the blows of wind are trying to equalize the horizontal imbalance in the air pressure. WDO is a new type nature-inspired global optimization based on atmospheric motion developed by Bayraktar et al. [153] in 2013. This method is working on the population-based iterative heuristic global optimization algorithm for multi-dimensional and multi-modal problems with the potential to implement constraints on the search domain. WDO is similar to other nature-inspired optimization algorithms, in which population-based heuristic iterative process can be found for solving multi-dimensional optimization problems [157]. At its center, a population of infinitesimally small air parcels navigates over an N-dimensional search space, employing Newton's second law of motion that is used to express the motion of air parcels inside the earth's atmosphere. As compared to similar particle based optimization algorithm (e.g., PSO), the WDO algorithm has additional terms in the velocity update equation such as Gravitation and Coriolis forces, which provides robustness and extra degrees of freedom to the algorithm.

Along with the theory and terminology of WDO, a numerical study for tuning the WDO parameters has been introduced in Bayraktar et al. [153]. WDO is further applied to different applications of electromagnetic optimization problems [153]. These papers signify that WDO can, in some cases, gives better performance in comparison to other well-known techniques like PSO, GA, and other optimization algorithms. The WDO algorithm is working based on the atmospheric motion of infinitesimal small air parcels navigates over an N-dimensional search domain. The starting step of this algorithm is supported by the Newton's second law of motion, which provides accurate results when applied to the analysis of atmospheric motion. It states that the total force applied on an air parcel causes it to accelerate with an acceleration a in the same direction as the applied total force.

$$\rho \times a = \sum F_i \tag{7.1}$$

where ρ is the density of air for an infinitesimally small air parcel, and F_i represents all the individual forces acting on the air parcel. To relate the air pressure to the air parcel's density and temperature, the ideal gas law can be utilized and is given by: -

$$P = \rho RT \tag{7.2}$$

where P is the pressure, R is the universal gas constant, and T is the temperature.

Four major forces can be included in the equation (7.1) that either cause the wind to move in a certain direction at a certain velocity or that deflect it from its existing path. The most observable force causing the air to move is the Pressure Gradient force F_{PG} defined in equation (7.3). Another force is the Friction force F_F defined in equation (7.4), which simply acts to oppose the motion started by the Pressure Gradient force. In the three-dimensional physical atmosphere, the Gravitational force F_G in equation (7.5) is a vertical force directed toward the earth's surface. The Coriolis force F_C in equation (7.6) is caused by due to the rotation of the earth and deflects the path of wind from one dimension to another.

$$F_{PG} = -\nabla P \times \delta V \tag{7.3}$$

$$F_F = -\rho \times \alpha \times u \tag{7.4}$$

$$F_G = \rho \times \delta V \times g \tag{7.5}$$

$$F_C = -2 \times \Omega \times u \tag{7.6}$$

where, ∇P is the pressure gradient, δV represents the infinite air volume, Ω represents the rotation of the earth, g is the gravitational acceleration, α is the friction coefficient and u is the velocity vector of the wind.

The sum of all forces (Gravitational force, Pressure Gradient force, Friction force, and Coriolis force) described above can be entered on the right-hand side of Newton's second law of motion given in equation (7.1), which leads to: -

$$\rho \times \frac{\Delta u}{\Delta t} = (\rho \times \delta V \times g) + (-\nabla P \times \delta V) + (-\rho \times \alpha \times u) + (-2 \times \Omega \times u) \qquad (7.7)$$

where the acceleration term in equation (7.1) is rewritten as $a = \Delta u / \Delta t$, and a time step $\Delta t = 1$ is assumed for simplicity. For an infinitesimally small, dimensionless air parcel, the volume is set as $\delta V = 1$, which simplifies the equation (7.7) to: -

$$\rho \times \Delta u = (\rho \times g) + (-\nabla P) + (-\rho \times \alpha \times u) + (-2 \times \Omega \times u) \qquad (7.8)$$

Putting the ideal gas law equation (7.2) in equation (7.8), the density ρ can be written in terms of the pressure, with temperature and the universal gas law constant: -

$$u_{new} = (1-\alpha) \times u_{cur} - g \times x_{cur} + \left(RT \left| \frac{1}{i} - 1 \right| (x_{opt} - x_{cur}) \right) + \left(\frac{c \times u_{cur}^{\text{other dim}}}{i} \right) \qquad (7.9)$$

where u_{new} is the velocity in the next iteration, u_{cur} is the velocity in current iteration, x_{cur} is the current location of the air parcel, x_{opt} is the optimum location of the air parcel, i represents the ranking between all air parcels, $u_{cur}^{\text{other dim}}$ is the velocity influence from another randomly chosen dimension of the same air parcel, and all other coefficients are combined into a single term c (i.e., $c = -2 \times \Omega \times RT$). Equation (7.9) represents the final form of the velocity update utilized in WDO [153, 155]. The following function updates the position of the air parcel: -

$$x_{new} = x_{cur} + (u_{new} \cdot \Delta t) \qquad (7.10)$$

where x_{new} is the new position of the air parcel in the next iteration. If the new velocity u_{new} exceeds the initialize maximum velocity ($u_{\max}$ = 0.3) in any dimension, then the velocity in that dimension is limited according to the following condition: -

$$u_{new}^{*} = \begin{cases} u_{\max} & \text{if } u_{new} > u_{\max} \\ -u_{\max} & \text{if } u_{new} < -u_{\max} \end{cases} \qquad (7.11)$$

where the direction of motion is preserved but the magnitude is limited to be no more than $|u_{\max}|$ at any dimension and u_{new}^{*} represents the adjusted velocity after it is limited to the maximum speed.

The pseudo-code of the WDO algorithm can be summarized as follows: -

Step 1. Start.

Step 2. Initialize the population size (i.e., group of air parcels), number of dimensions of the optimization problem, maximum number of iterations, coefficients (such as RT, g, α, c, $u_{\max}$), pressure function (fitness function of the optimization problem), lower and upper boundaries of the optimization problem.

Step 3. Assign random position and velocity of the air parcels.

Step 4. Evaluate the pressure (fitness) values of each air parcel at its current position.

Step 5. Once the pressure values have been evaluated, the population is ranked based on their pressure (ascending order), and the velocity updated according to equation (7.9) along with the restrictions are given in equation (7.11).

Step 6. Update the position of the air parcel for the next iteration according to equation (7.10) and also check the boundaries of the air parcel.

Step 7. Stop if a maximum number of iterations are achieved, else go to step 4.

When the maximum number of iterations is completed, the best pressure (objective) value is achieved.

In the Wind driven optimization (WDO) based path planning problems, the objective function is considered as the best steering angle relies on the criterion of the random search model. This optimum steering angle is helping to achieve the minimum path length in the given environment. The objective function measures the optimum solution of the current model. In this chapter, the objective function values are Front Obstacle Distance (F.O.D.), Right Obstacle Distance (R.O.D.), and Left Obstacle Distance (L.O.D.). Obstacle distance is used to optimize the following objective equation (7.12) for getting optimum Steering Angle (S.A.) for the proposed model:-

$$S.A. = -11.217 + 0.0016 * F.O.D. - 0.7326 * R.O.D. + 1.1317 * L.O.D. \tag{7.12}$$

where,

$20 \le F.O.D. \le 150$ **(Ranging between 20 to 150 same as sensor data range)**

$20 \le R.O.D. \le 150$ **(Ranging between 20 to 150 same as sensor data range)**

$20 \le L.O.D. \le 150$ **(Ranging between 20 to 150 same as sensor data range)**

The selection of the objective function is based on two reasons, i.e. obstacle avoidance and find out the shortest possible path in an environment. These obstacle distances (F.O.D., R.O.D., and L.O.D.) are acquired by the different equipped sensors such as ultrasonic range finder sensor, sharp infrared range sensor, and other sensors. The sensing range of sharp infrared range sensor is between 20cm to 150cm, and the ultrasonic range finder sensor is 2cm to 4m. Both sensors are used for sensing the front, left, and right placed obstacles around the robot. If the robot moves in an environment filled with static and dynamic obstacles, it is necessary to detect the obstacles and avoid it. The developed architecture of mobile robot navigation based on WDO method is given in Figure 7.1.

7.3 Computer Simulation Results and Discussion

In order to verify the effectiveness of the proposed WDO algorithm, the different simulations have been done using MATLAB software. The WDO algorithm ran with the different population size of 20, 30, 40 air parcels, and three number of dimensions for a maximum of 500 epochs (iterations). Other necessary parameters used for WDO, GA, and PSO algorithm are listed in Tables 7.1 to 7.3, respectively. Simulation results in Figures 7.2 to 7.6 demonstrate the mobile robot navigation and obstacle avoidance in the different static and dynamic environments. It is assumed that the position of the start point and goal point are known. But the positions of all the obstacles in the environment are unknown for the robot. The obstacle distance information obtained by the equipped sensors are the inputs of the navigational controller, which provides the necessary optimum steering angle control command as its output for navigation. In order to demonstrate the efficiency of this new optimization technique, it is applied to different unknown simulation environments and compared with other optimization techniques like Genetic Algorithm, Particle Swarm Optimization and the results are found to be in agreement in terms of navigation path length and also the time taken to reach the target (Tables 7.4 and 7.5) and are illustrated in Figures 7.7 to 7.8. The average deviation path length between WDO versus GA is 0.7%, and the WDO versus PSO is 1.640%.

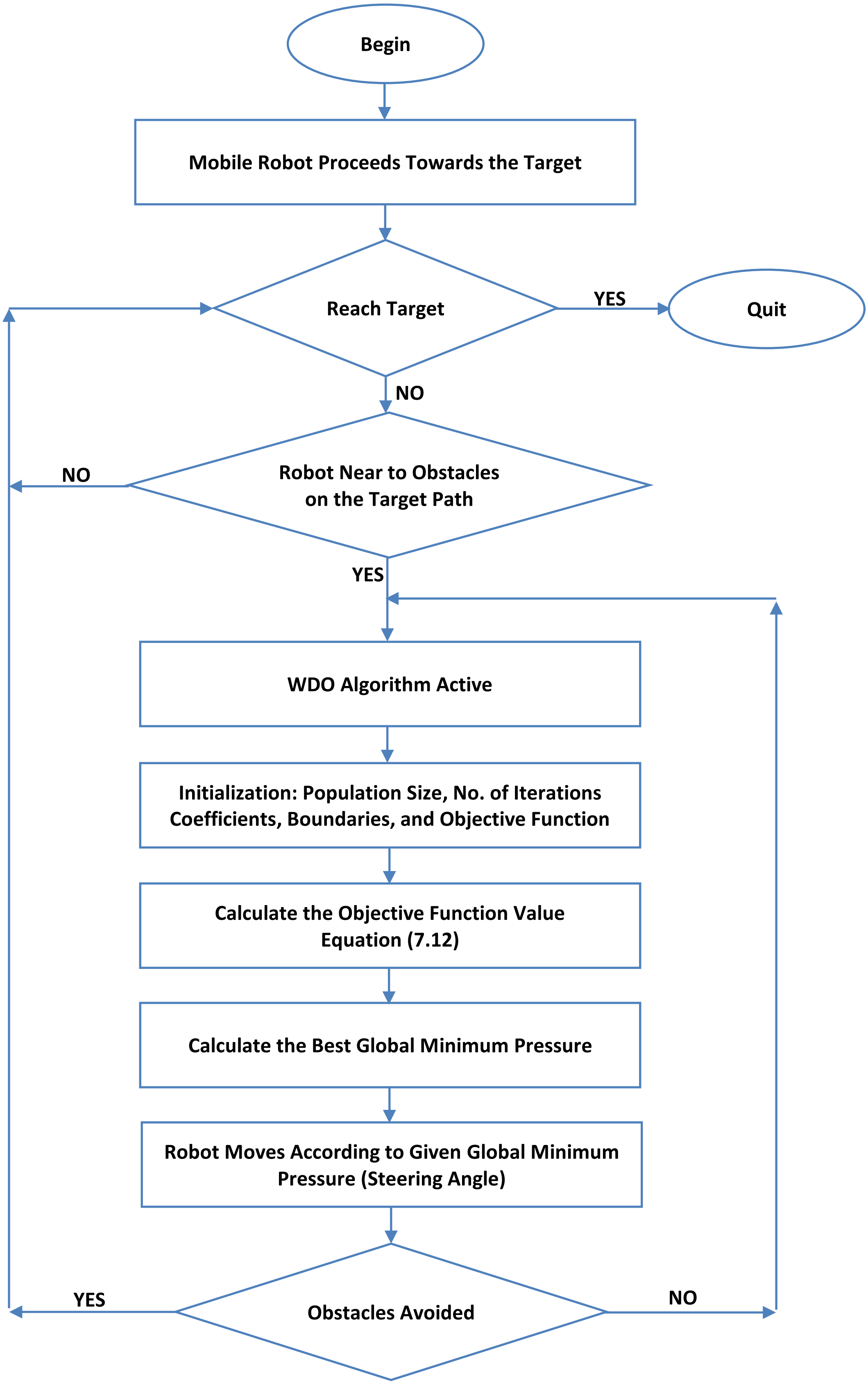


Figure 7.1: The architecture of mobile robot navigation based on WDO method.

Table 7.1: Parameters used in WDO algorithm

S. No.	Parameters	Values
1.	Population size	20
2.	Dimension of the problem	3
3.	Maximum number of iterations	500
4.	RT coefficient	3
5.	Gravitational constant	0.2
6.	Friction coefficient	0.4
7.	Coriolis effect	0.4
8.	Velocity limit	0.3

Table 7.2: Parameters used in GA

Sl. No.	Parameters	Values/Function Name
1.	Population size	20
2.	Selection function	Stochastic uniform
3.	Elite count	1.8
4.	Crossover fraction	0.7
5.	Mutation function	Constraint dependent
6.	Crossover function	Scattered
7.	Number of generation	100

Table 7.3: Parameters used in PSO algorithm

Sl. No.	Parameters	Values
1.	Swarm size	150
2.	Maximum number of iterations	400
3.	Social acceleration factors	2
4.	Cognitive acceleration factors	2

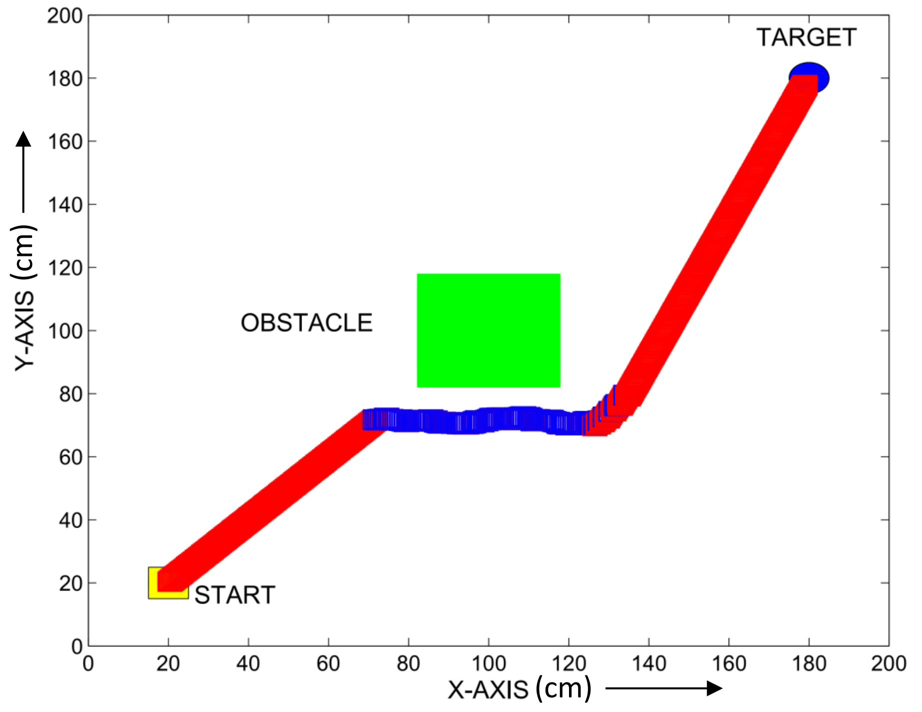


Figure 7.2: Navigation of a mobile robot using WDO technique.

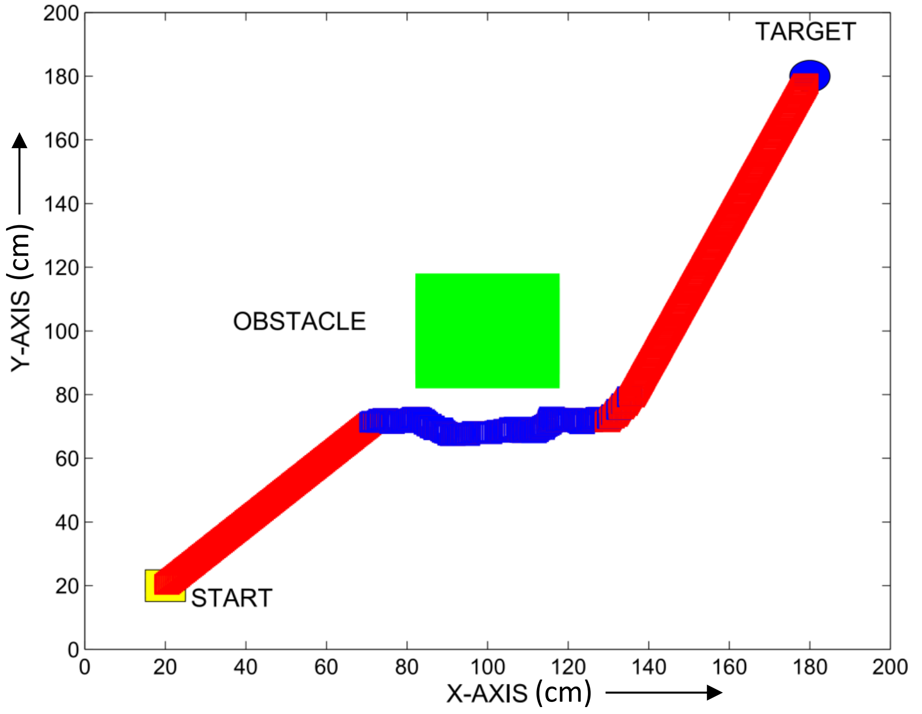


Figure 7.3: Navigation of a mobile robot using GA technique.

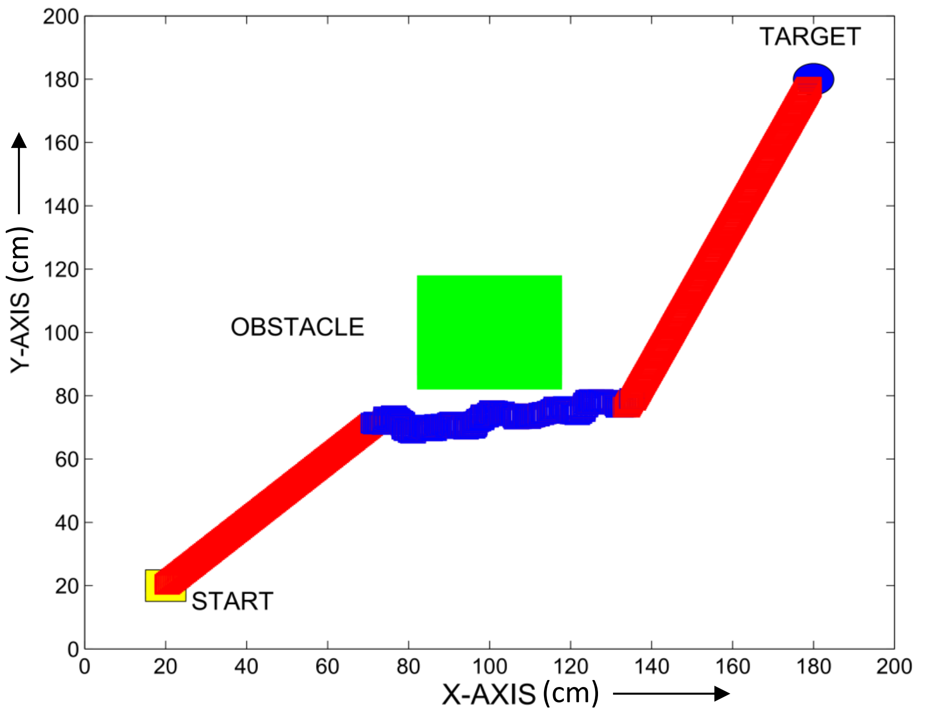


Figure 7.4: Navigation of a mobile robot using PSO technique.

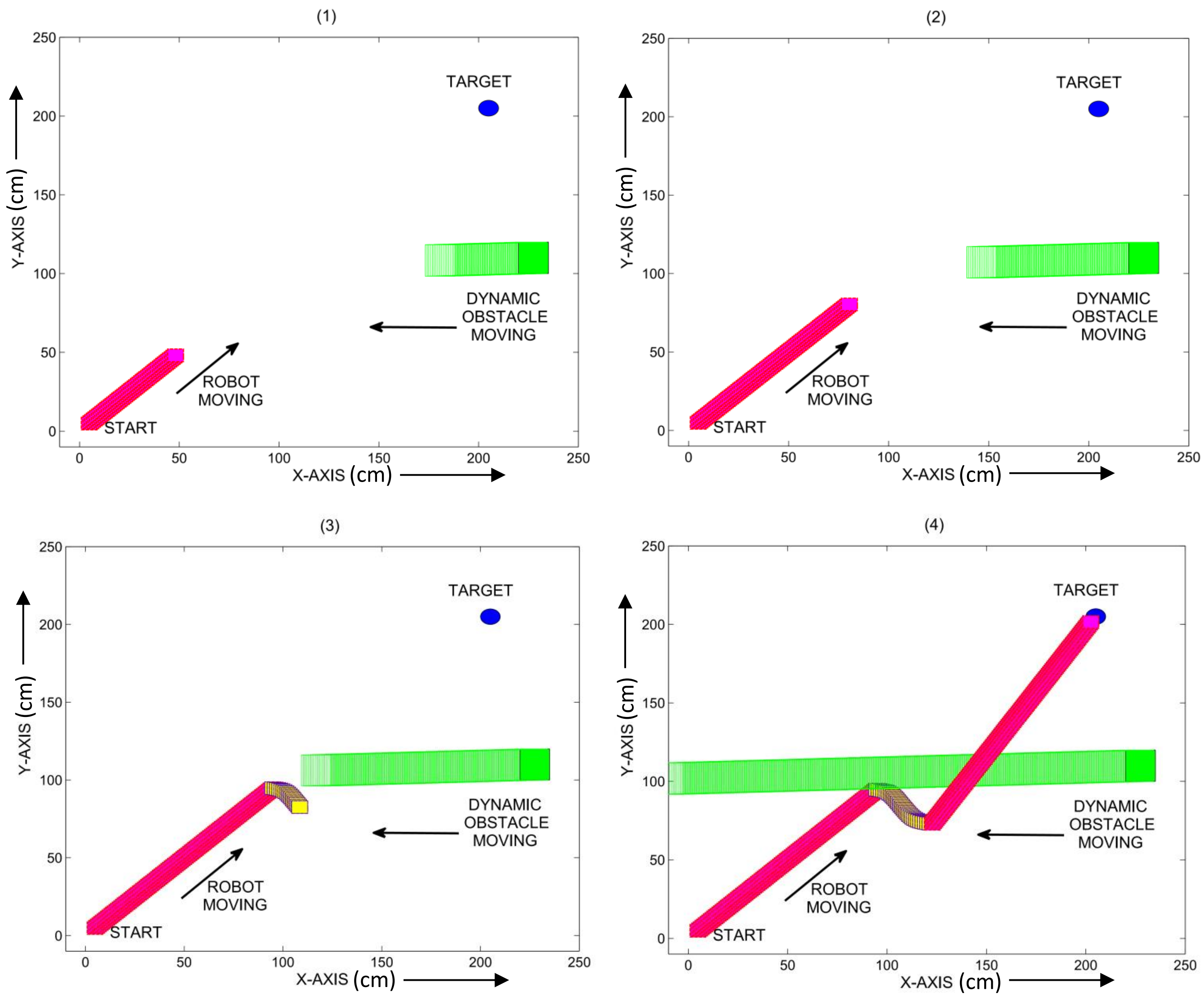


Figure 7.5: Navigation of a mobile robot using WDO technique in a dynamic environment.

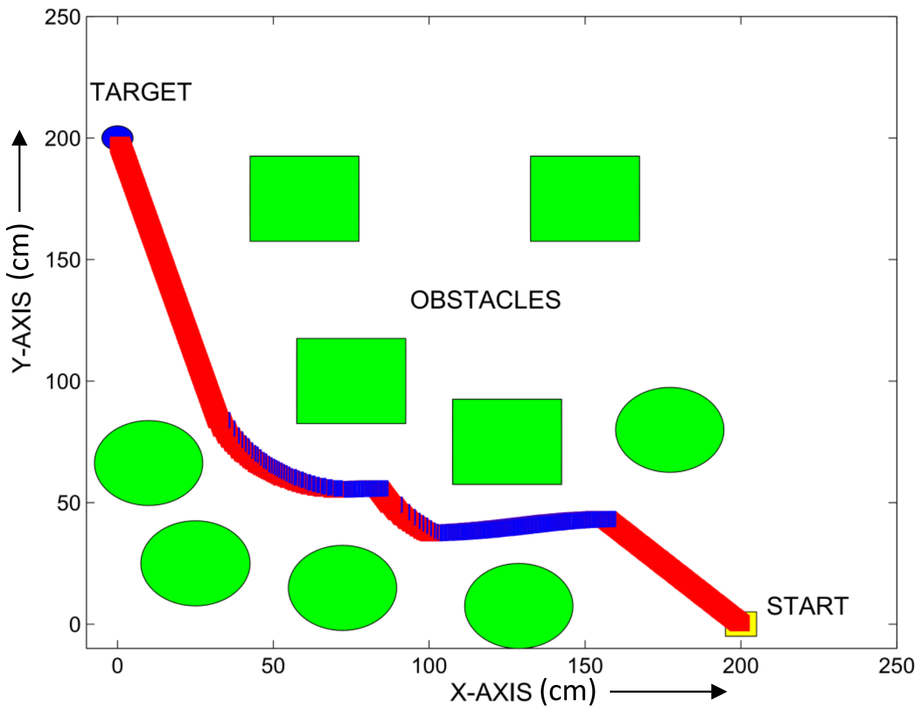


Figure 7.6: Navigation of a mobile robot using WDO technique in a cluttered environment.

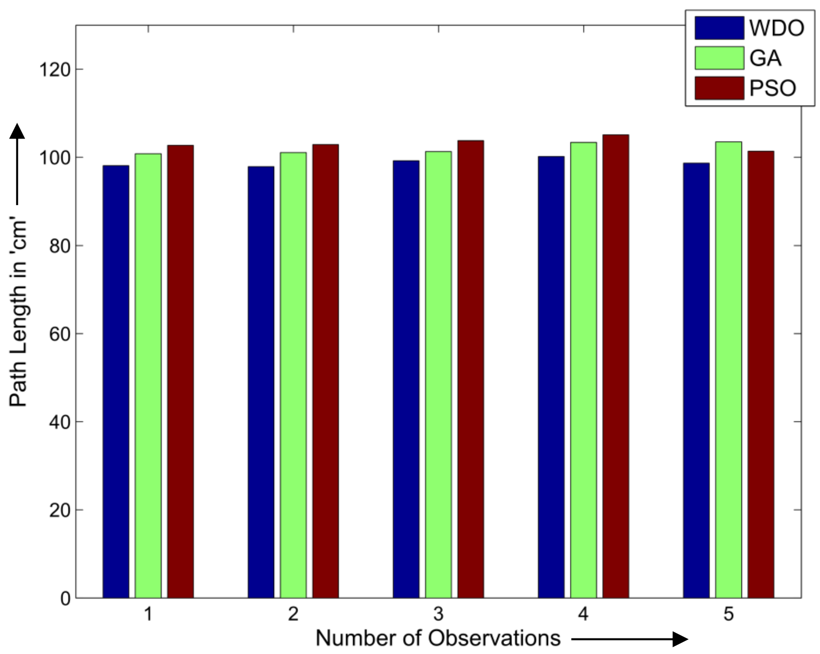


Figure 7.7: Comparison performance graph between WDO algorithm over GA, and PSO in terms of navigation path length.

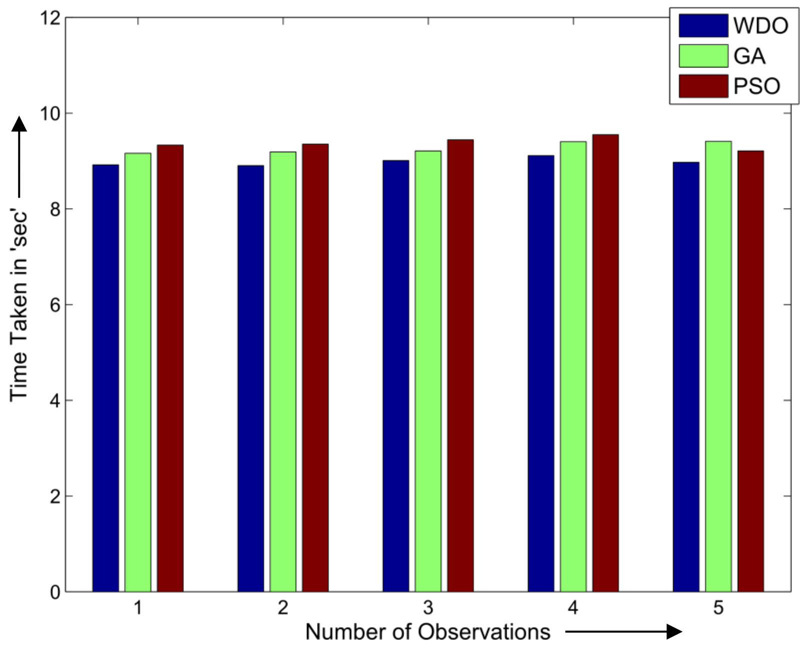


Figure 7.8: Comparison performance graph between WDO algorithm over GA, and PSO in terms of time taken to reach the target.

Table 7.4: Comparison the performance of WDO algorithm over GA and PSO in terms of navigation path length

Navigation path length (cm) of the a robot using WDO in Figure 7.2	**Navigation path length (cm) of the a robot using GA in Figure 7.3**	**Navigation path length (cm) of the a robot using PSO in Figure 7.4**
98	100	103
97	**101**	**102**
99	101	103
100	103	105
98	104	101

Note: Bold value indicates the minimum path length.

Table 7.5: Comparison the performance of WDO algorithm over GA and PSO in terms of time taken to reach the target

Time taken (sec) to reach the target by the robot using WDO in Figure 7.2	Time taken (sec) to reach the target by the robot using GA in Figure 7.3	Time taken (sec) to reach the target by the robot using PSO in Figure 7.4
8.92	9.16	9.33
8.9	**9.19**	**9.35**
9.01	9.21	9.44
9.11	9.4	9.55
8.97	9.41	9.21

Note: Bold value indicates the minimum time.

7.4 Comparison with Previous Navigational Controllers

A comparison has been done on path generated by the Genetic Algorithm [169], and Particle Swarm Optimization [173] over the current Wind Driven Optimization algorithm based navigation technique in both simulation and experimental modes. The performance of this method is evaluated on the basis of navigation path length.

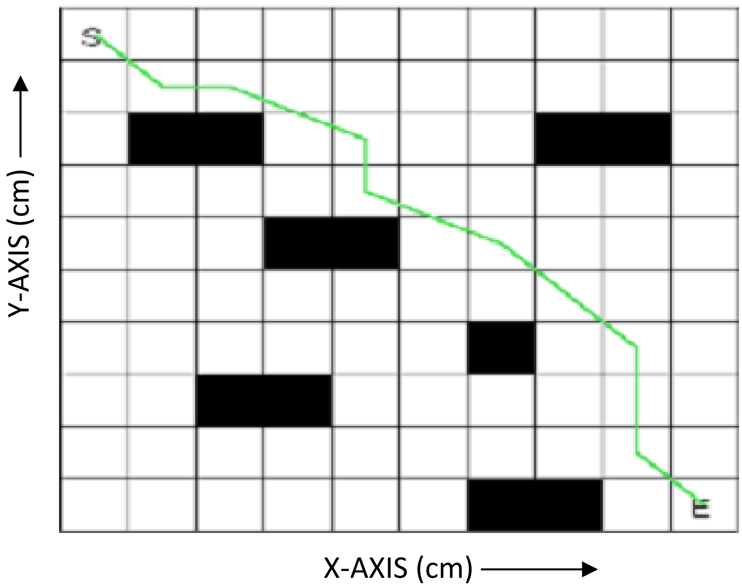
Jianguo et al. [169] have proposed an adaptive Genetic Algorithm (GA) method that automatically adjusts the GA parameters like crossover and mutation, according to the different changeable environment for a mobile robot. They also compared this adaptive Genetic Algorithm method to traditional GA and stated that the adaptive Genetic Algorithm performs better over traditional GA.

Deepak et al. [173] have presented Particle Swarm Optimization based technique that avoids the obstacle efficiently and generates a feasible path in unknown environments. They performed various graphical simulation tests in a different scenario to verify the effectiveness of the developed technique.

Figures 7.9 (i) and 7.9 (ii) show the comparison between path generated by GA [169] and the current WDO algorithm for the same start position to target position. WDO generated an optimum path in comparison to GA for the same path planning problem. Similarly, Figures 7.10 (i) and 7.10 (ii) illustrate the comparison between path generated by PSO [173] and the current WDO algorithm for the same start position to target position. WDO gives an optimum path in comparison to PSO for the same path planning problem.

In Table 7.6, GA [169] based navigational controlled robot finds the target with the path length of 115 cm and the proposed WDO algorithm based navigational controlled robot finds the target with an optimum path length of 111 cm. Similarly, in Table 7.7, PSO [173] based navigational controlled robot finds the target with the path length of 171 cm and the proposed WDO algorithm based navigational controlled robot finds the target with optimal path length of 160 cm. During comparison (Tables 7.6-7.7), it can be seen that the proposed controller gives better results (in terms of path length and smoothness) compared to the previous navigational controller [169, 173]. The centimeter measurements are taken on the proportional basis.

(i) GA



(ii) WDO

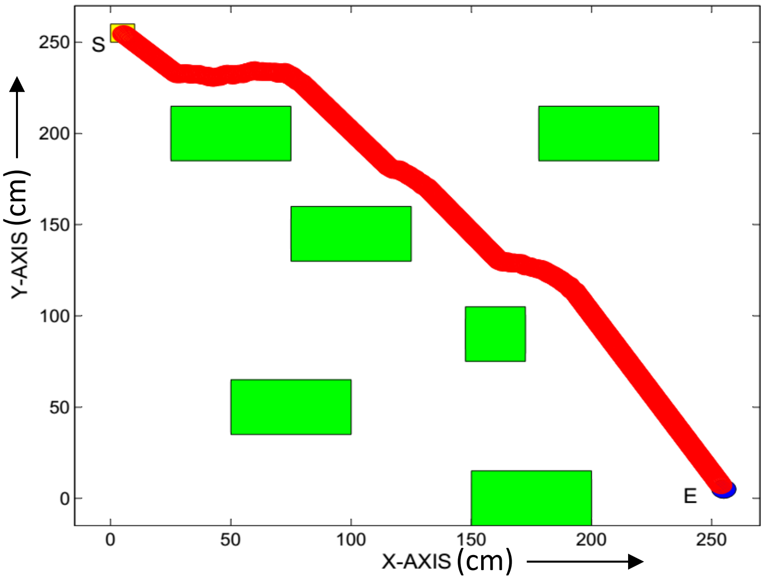
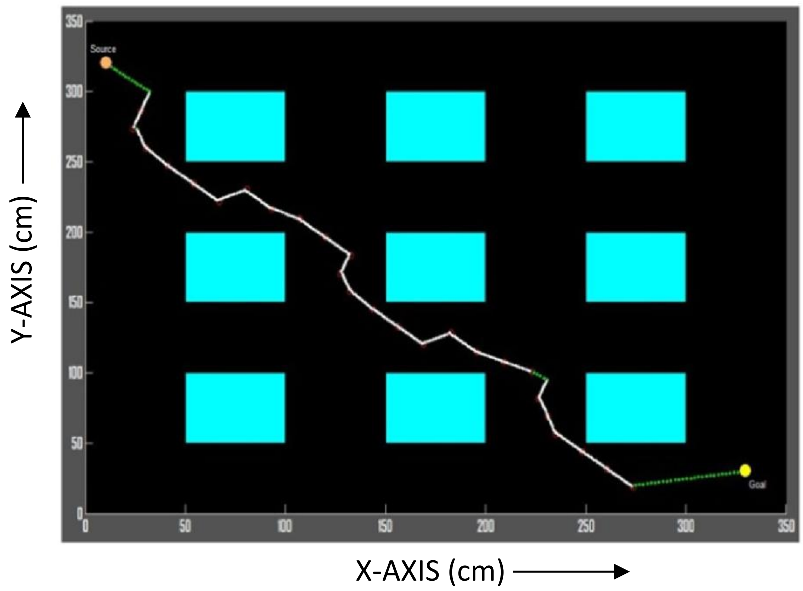


Figure 7.9: A simulation comparison results between GA (i) and WDO (ii).

Table 7.6: Results of Jianguo et al. [169] method and WDO algorithm

Figure no.	GA based path length (cm) of Jianguo et al. [169]	WDO algorithm based path length (cm)
Figure 7. 9 (i) and (ii)	115	111

(i) PSO



(ii) WDO

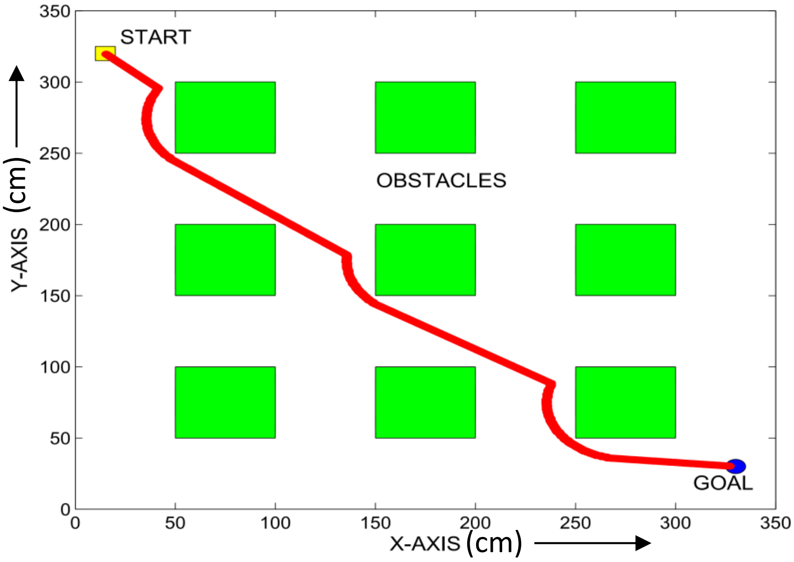


Figure 7.10: A simulation comparison results between PSO (i) and WDO (ii).

Table 7.7: The comparison result between Deepak et al. [173] method and WDO algorithm

Figure no.	PSO based path length (cm) of Deepak et al. [173]	WDO algorithm based path length (cm)
Figure 7.10 (i), (ii)	171	160

7.5 Experimental Results and Discussion

In order to validate the effectiveness of the proposed WDO algorithm, real-time experiments are done on the real four-wheeled mobile robot (Figure 7.11) in unknown environments. The main specifications of the proposed prototype experimental mobile robot are given in Table 7.8. The WDO based navigational controller is implemented in the Arduino MEGA (ATmega2560) microcontroller. The experimental prototype of four-wheeled differentially steer controlled mobile robot is shown in Figure 7.11. The motion and orientation are controlled by independent four DC geared motors, which provide the necessary torque to all driving wheels. The orientation of the mobile robot is achieved by the differential steered control (motor speed) of all wheels. Four separate 12Volt DC motors are attached to a dual DC motor driver (L298) and the each driver's direction and velocity control pins are connected to the Arduino MEGA microcontroller to drive each motor to facilitate turn left and right, backward and forward movements, and the motor velocity regulated by the Pulse Width Modulation (PWM) signal. The mobile robot uses two kinds of sensors such as two ultrasonic range finder (HC SR-04) sensors and one sharp infrared range sensor (GP2Y0A02YK0F) shown in Figure 7.12 (Schematic diagram of differentially steered four-wheeled mobile robot). The sensors are used to prevent a collision from surrounding obstacles. To sense the obstacles one sharp infrared sensor is mounted on the front side of the robot for front obstacle detection, and two ultrasonic range finder sensors are attached to the left and right sides of the robot for left and right obstacle detection respectively. The two groups of ultrasonic range finder sensors mounted on the left, and right side of the robot is connected to the digital input port of the Arduino microcontroller. The single sharp infrared range sensor mounted on the front side of the robot is connected to the analog input port of the Arduino MEGA microcontroller.

Experimental verification of the above simulation results has been shown in Figures 7.13 and 7.14. In the real-time experimental result analysis, it can be indicated that the mobile robot reaches their target position successfully without hitting any obstacles in the given cluttered environment. It is assumed that the position of the start point and goal point are known, but the positions of all the obstacles in the environment are unknown for the robot. These experimental results depict that the mobile robot can find a collision-free near-optimal path in unknown environments using on-board sensory information. Figure 7.15 illustrates the experimentally obtained navigation paths follow closely those traced by the robot during simulation results. Table 7.9 shows the real-time navigation path length and time taken by the robot in the various unknown environments. Tables 7.10 and 7.11 illustrate the travelling path length and navigation time comparison between the simulation and experimental results. In the comparison study between the simulation and experiments, it is observed that some errors have been found, these are happened due to slippage and friction during real time experiment.

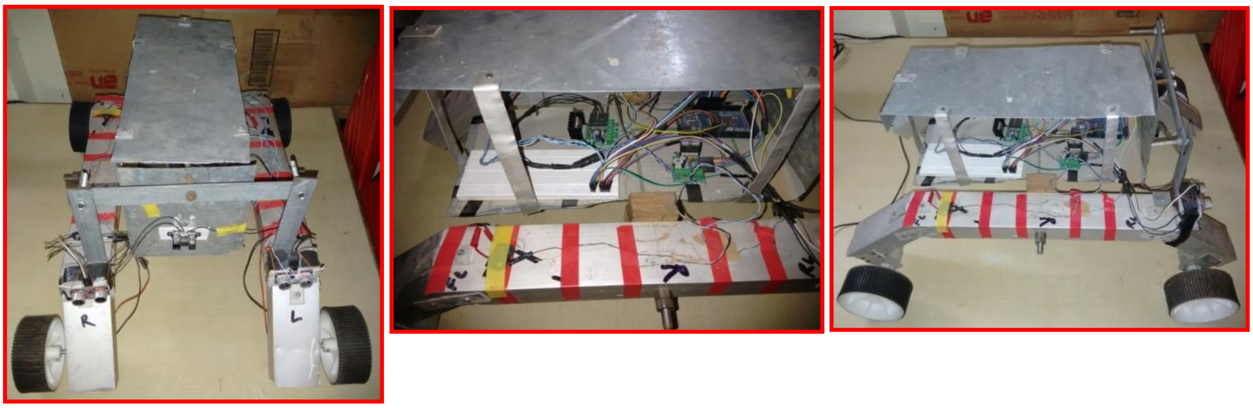


Figure 7.11: Experimental four-wheeled real mobile robot.

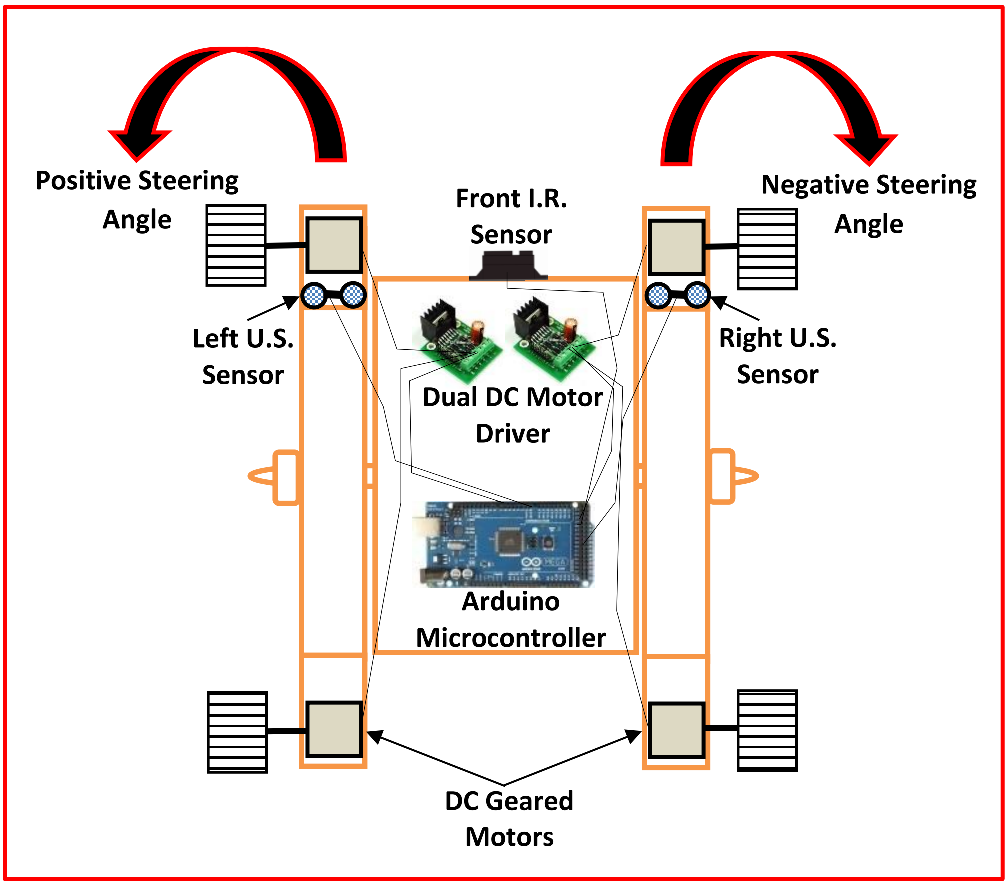


Figure 7.12: Schematic diagram of differentially steered four-wheeled mobile robot.

Table 7.8: Main specifications of the proposed prototype experimental mobile robot

Name	**Specifications**
Microcontroller	Arduino MEGA 2560 (ATmega2560).
Flash Memory	256 KB (ATmega2560).
Operating Voltage	5V.
SRAM	8 KB (ATmega2560).
Input Voltage	7–12V (Recommended).
Input Voltage (Limits)	6–20V.
Digital Input/Output Pins	54 (of Which 15 Can be Used as PWM Outputs).
Analog Input Pins	16.
Motors	4 DC, 30RPM Centre Shaft Economy Series DC Motor.
Motors Driver	L298, Up to 46V, 2A Dual DC Two Motor Drivers.
Motor Speed	Max: 30RPM, Min: 12RPM.
Wheel	Wheel Diameter: 106mm, Wheel Thickness: 44mm, Hole Diameter: 8mm.
Sensors	One Sharp Infrared Range Sensor (GP2Y0A02YK0F) Distance Measuring Range: 20cm to 150cm.
	Two Ultrasonic Range Finder Sensor (HC SR-04) Distance Measuring Range: 2cm to 400cm.
Bread Board	Small Size Bread Board.
Communication	USB connection Serial Port.
Principal Dimensions	Height: 30cm, Length: 62cm, Width: 49cm.
Weight	Approx. 4.936kg.
Payload	Approx. 500g.
Power	Two Rechargeable Lithium Polymer 3 Cell, 11.1V, 2000mAh, 20C Battery

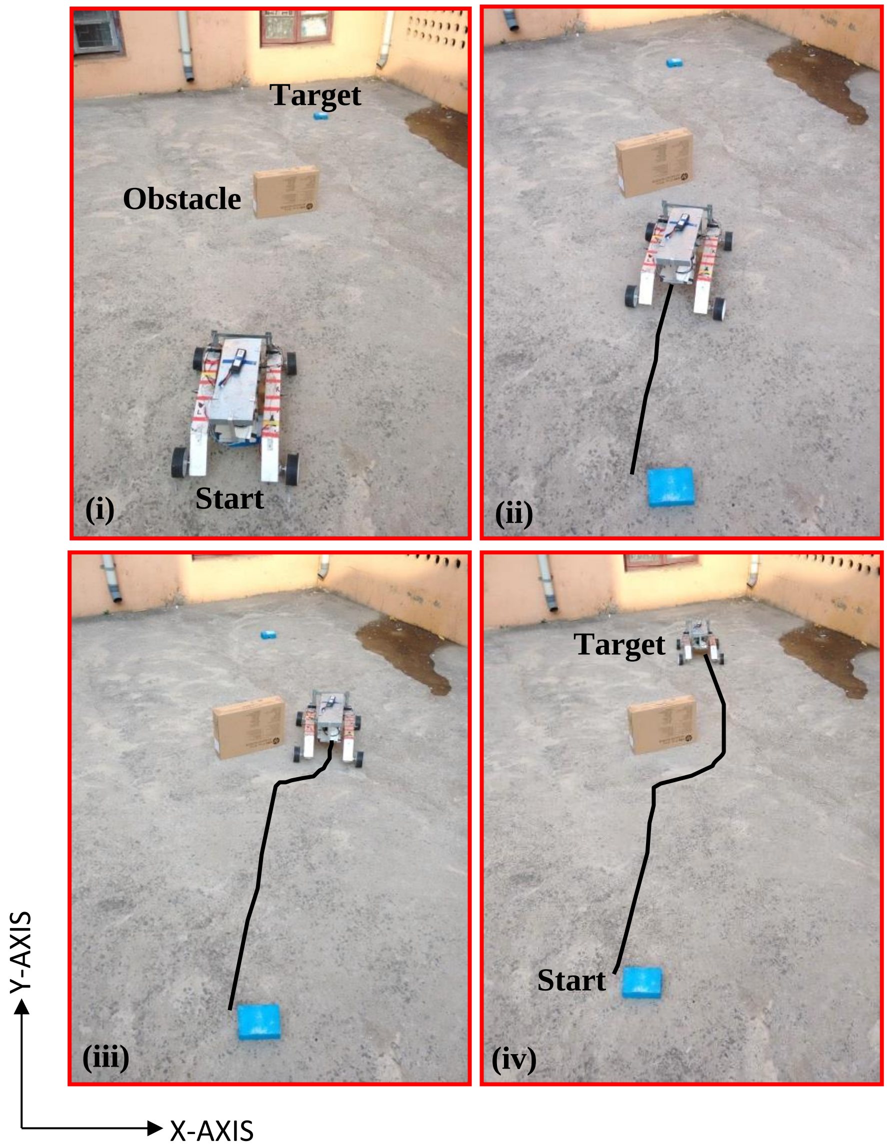


Figure 7.13: Experimental result of mobile robot navigation same as a simulation environment (shown in Figure 7.2).

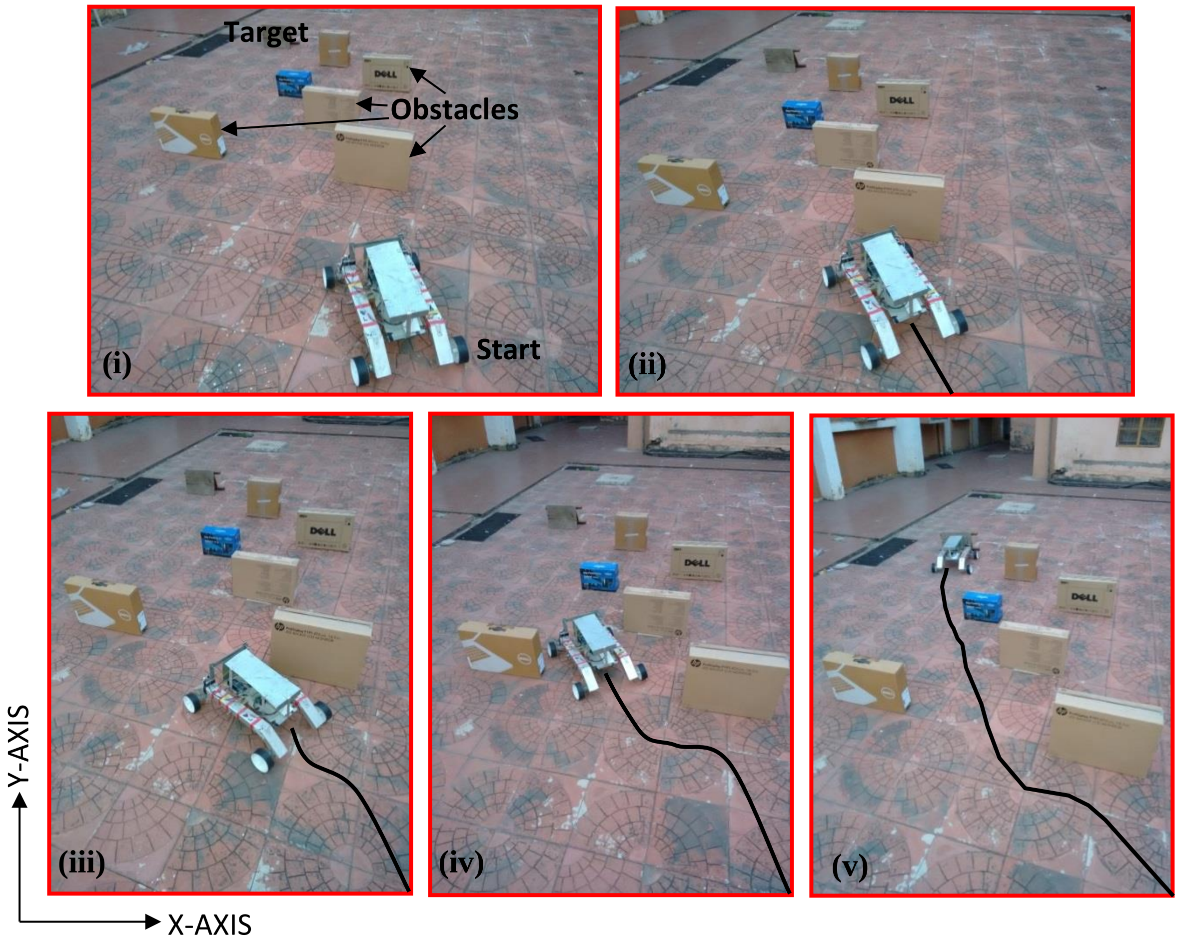


Figure 7.14: Experimental result of mobile robot navigation same as a simulation environment (shown in Figure 7.9 (ii)).

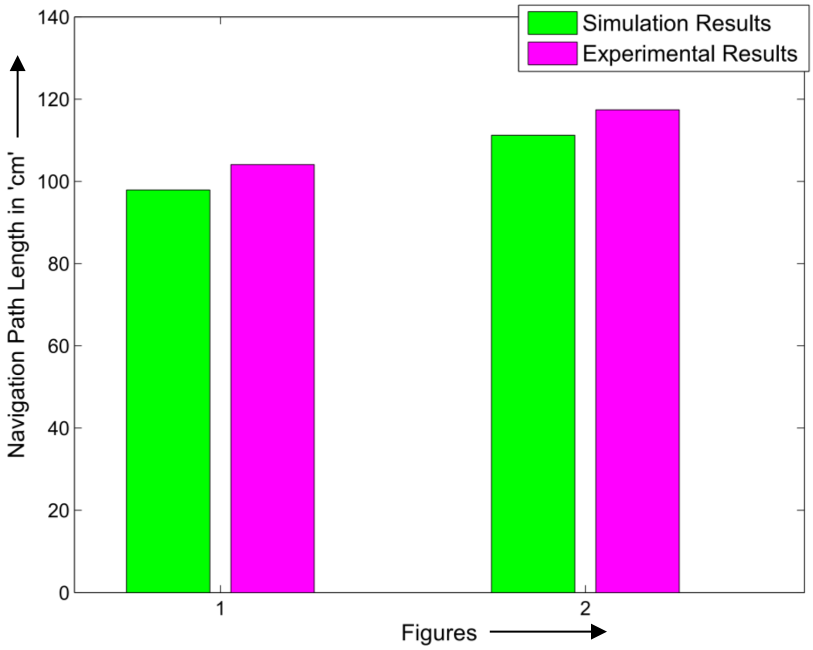


Figure 7.15: Comparison of path length between simulation and experimental results.

Table 7.9: Experimental results of mobile robot navigation in the different environments using WDO algorithm

Figure no.	**Environment type**	**Experimental path length (cm)**	**Navigation time (sec)**
Figure 7.13	Single obstacle environment	103	9.5
Figure 7.14	Cluttered environment	117	10.3

Table 7.10: Navigation path lengths between simulation and experimental results

Figure no. (Simulation and experimental res.)	**Navigation path length (cm)**		**Error between simulation and experimental result**
	Simulation result	**Experimental result**	
Figures 7.2 and 7.13	97	103	5.82%
Figures 7.9 (ii) and 7.14	111	117	5.13%

Table 7.11: Navigation time comparison between simulation and experimental results

Figure no. (Simulation and experimental res.)	**Navigation time (sec)**		**Error between simulation and experimental result**
	Simulation result	**Experimental result**	
Figures 7.2 and 7.13	8.9	9.5	6.32%
Figures 7.9 (ii) and 7.14	9.6	10.3	6.79%

7.6 Summary

In order to demonstrate the success of this new optimization technique, it is applied to the different unknown cluttered simulation and experimental environments. Comparisons have been done with previous techniques like adaptive Genetic Algorithm [169], Particle Swarm Optimization [173] and are found to be in agreement in terms of navigation path length. The Wind Driven Optimization based path planning for a mobile robot finds an optimum or near-optimum path from a start point to target point in the static and dynamic environments filled with obstacles. The simulation and the experimental results show the proposed optimization algorithm for a mobile robot navigation problem is feasible and practical. Using this algorithm, the robot can also negotiate with dynamic obstacles successfully. In the comparison study between the simulation and experiment results errors are recorded, and the errors are found due to the effect of slippage and friction between the wheels of the robot and surface during navigation in real time mode. During experiment utmost care has been taken to minimize the slippage and friction between the wheels and surface. Still the effect of slippage and friction are unavoidable, and errors are recorded during the comparison of the results for travelling path length (5.48%) and for navigation time (6.56%). In future, this technique can be hybridized with another algorithm to improve the navigational performance of the mobile robot.

Chapter 8

Optimum Path Planning of Mobile Robot in Unknown Static and Dynamic Environments using Fuzzy-Wind Driven Optimization Algorithm

8.1 Introduction

This chapter introduces a singleton fuzzy (S-Fuzzy) controller and Fuzzy-WDO hybrid algorithm for the autonomous mobile robot navigation and collision avoidance in an unknown static and dynamic environment. The WDO (Wind Driven Optimization) algorithm is used to optimize and tune the input/output membership function parameters of the fuzzy controller. The WDO algorithm is working based on the atmospheric motion of infinitesimal small air parcels navigates over an N-dimensional search domain. The performance of this proposed technique has compared through many computer simulations and real-time experiments by using Khepera-III mobile robot. As compared to the S-Fuzzy controller the Fuzzy-WDO algorithm is found good agreement for mobile robot navigation.

One major problem with the fuzzy logic is the difficulty of constructing and tuning the correct membership function grade [174]. Therefore, the authors have tried to solve this problem by using WDO algorithm. In this chapter, a fuzzy-WDO hybrid algorithm has been presented for mobile robot navigation and collision avoidance in an unknown static and dynamic environment. The WDO is integrated with the fuzzy controller to adjust and optimize the antecedent and consequent parameters of the generalized bell-shaped membership function. The WDO [153] method is a population-based iterative heuristic global optimization algorithm for multi-dimensional and multi-model problems with the potential to implement constraints on the search domain. This algorithm works by simultaneously maintaining several infinitesimal small air parcels or potential solutions in the search domain. For each iteration of the algorithm, each air parcels are evaluated by the membership function parameters (objective function) being optimized based on the fitness function of that solution. The primary objective of this research is to optimize the membership function parameters of the fuzzy controller by using WDO algorithm.

This chapter is organized into seven sections. Section 8.1 presents the introduction. Singleton fuzzy (S-Fuzzy) controller for mobile robot navigation is proposed in Section 8.2. The hybrid fuzzy-WDO algorithm for mobile robot navigation is presented in Section 8.3. Section 8.4 demonstrates the simulation results of the mobile robot in different environments. Section 8.5 describes the simulation result comparison with previous works. Section 8.6 presents the experimental results and discussion for validating the proposed controller. Finally, Section 8.7 depicts the summary.

8.2 Singleton Fuzzy (S-Fuzzy) Controller for the Mobile Robot Navigation

In this section, a singleton fuzzy (S-Fuzzy) rule-based controller has been designed and implemented for mobile robot navigation and collision avoidance in an unknown static and dynamic environment. The S-Fuzzy controller is used to control the right motor velocity and left motor velocity of the mobile robot. The S-Fuzzy controller has three inputs: Forward Obstacle Distance (d_f), Left Forward Obstacle Distance (d_l) and Right Forward Obstacle Distance (d_r); and two outputs: Right Motor Velocity (m_r) and Left Motor Velocity (m_l), which are logically connected by eight rules (see the Figure 8.2). These input and output variables are illustrated in Figures 8.3 and 8.4, respectively. The fuzzy rule set of the S-Fuzzy controller is described in Table 8.1. The two generalized bell-shaped (Gbell) membership functions are used for inputs and outputs. The range of inputs is divided into two linguistic variables: NEAR and FAR. These inputs are located at 20cm to 150cm. Similarly, the two Gbell membership functions (MFs) LOW and HIGH, respectively have been used for the outputs, and it is located at 6.7cm/sec to

16.7cm/sec. The S-Fuzzy controller is composed through Mamdani-type fuzzy model in the following form: -

$$Rule_n: \ IF \ d_f \ is \ d_{f(i)}, \ d_l \ is \ d_{l(j)}, \ \& \ d_r \ is \ d_{r(k)} \ THEN \ m_r \ is \ m_{r(ijk)} \ \& \ m_l \ is \ m_{l(ijk)} \tag{8.1}$$

where n=1, 2,…,8 (eight rules), the i=1, 2, j=1, 2 and k=1, 2 because d_f, d_l and d_r have two Gbell membership functions each. The $d_{f(i)}$, $d_{l(j)}$, and $d_{r(k)}$ are the fuzzy sets of the inputs d_f, d_l, and d_r, respectively. Similarly, the $m_{r(ijk)}$, and $m_{l(ijk)}$ are the fuzzy sets of the outputs m_r, and m_l, respectively. The fuzzy set (inputs and outputs) uses the following Gbell membership function: -

Let d_f, d_l, and d_r are presented by x_1, x_2, and x_3 respectively. Similarly, m_r, and m_l are denoted by y_1, and y_2, respectively.

$$\mu_{n1}(x_1) = \frac{1}{1+\left|\frac{x_1 - c_{n1}}{a_{n1}}\right|^{2b_{n1}}} \tag{8.2}$$

$$\mu_{n2}(x_2) = \frac{1}{1+\left|\frac{x_2 - c_{n2}}{a_{n2}}\right|^{2b_{n2}}} \tag{8.3}$$

$$\mu_{n3}(x_3) = \frac{1}{1+\left|\frac{x_3 - c_{n3}}{a_{n3}}\right|^{2b_{n3}}} \tag{8.4}$$

$$\mu_{n1}(y_1) = \frac{1}{1+\left|\frac{y_1 - c_{n1}}{a_{n1}}\right|^{2b_{n1}}} \tag{8.5}$$

$$\mu_{n2}(y_2) = \frac{1}{1+\left|\frac{y_2 - c_{n2}}{a_{n2}}\right|^{2b_{n2}}} \tag{8.6}$$

where a, b, and c are adjusting parameters of the membership function; called as the half width, slope control, and center respectively. The general structure of the generalized bell-shaped membership function is shown in Figure 8.1.

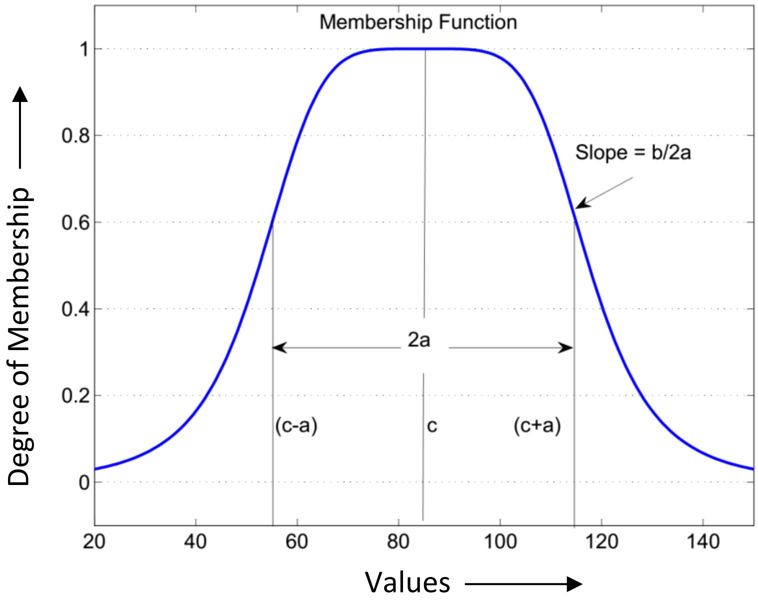


Figure 8.1: The general structure of the generalized bell-shaped membership function.

The defuzzification of the outputs (y_1 and y_2) are accomplished by the weighted average method: -

$$y_1 = \frac{\sum_{n=1}^{8}\left(\mu_{n1}(x_1)\cdot\mu_{n2}(x_2)\cdot\mu_{n3}(x_3)\right)\cdot y_1}{\sum_{n=1}^{8}\left(\mu_{n1}(x_1)\cdot\mu_{n2}(x_2)\cdot\mu_{n3}(x_3)\right)} \tag{8.7}$$

$$y_2 = \frac{\sum_{n=1}^{8}\left(\mu_{n1}(x_1)\cdot\mu_{n2}(x_2)\cdot\mu_{n3}(x_3)\right)\cdot y_2}{\sum_{n=1}^{8}\left(\mu_{n1}(x_1)\cdot\mu_{n2}(x_2)\cdot\mu_{n3}(x_3)\right)} \tag{8.8}$$

The adjusting parameters a, b, and c of the inputs and outputs are listed in Tables 8.2 and 8.3, respectively, which will be optimized through the WDO algorithm in Section 8.3 below.

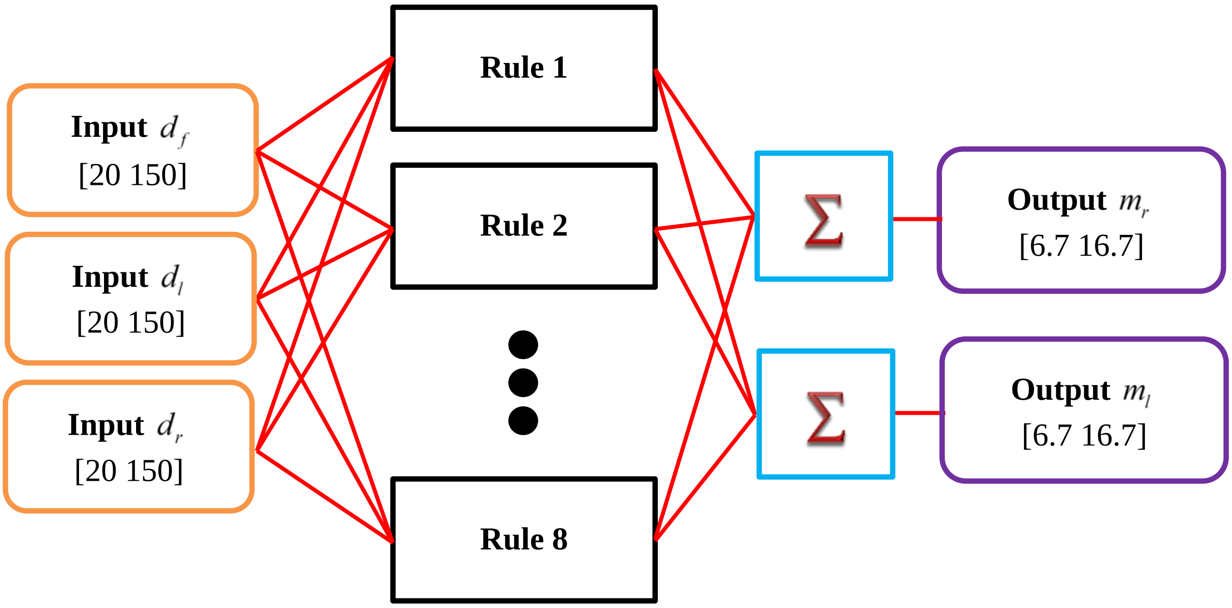


Figure 8.2: The structure of an S-Fuzzy controller for mobile robot navigation.

Table 8.1: Fuzzy rules set

If (d_f is Far) **and** (d_l is Far) **and** (d_r is Far) **then** (m_r is High) **and** (m_l is Low)
If (d_f is Near) **and** (d_l is Near) **and** (d_r is Near) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Far) **and** (d_l is Near) **and** (d_r is Far) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Far) **and** (d_l is Far) **and** (d_r is Near) **then** (m_r is High) **and** (m_l is Low)
If (d_f is Near) **and** (d_l is Far) **and** (d_r is Far) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Near) **and** (d_l is Near) **and** (d_r is Far) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Near) **and** (d_l is Far) **and** (d_r is Near) **then** (m_r is High) **and** (m_l is Low)
If (d_f is Far) **and** (d_l is Near) **and** (d_r is Near) **then** (m_r is Low) **and** (m_l is High)

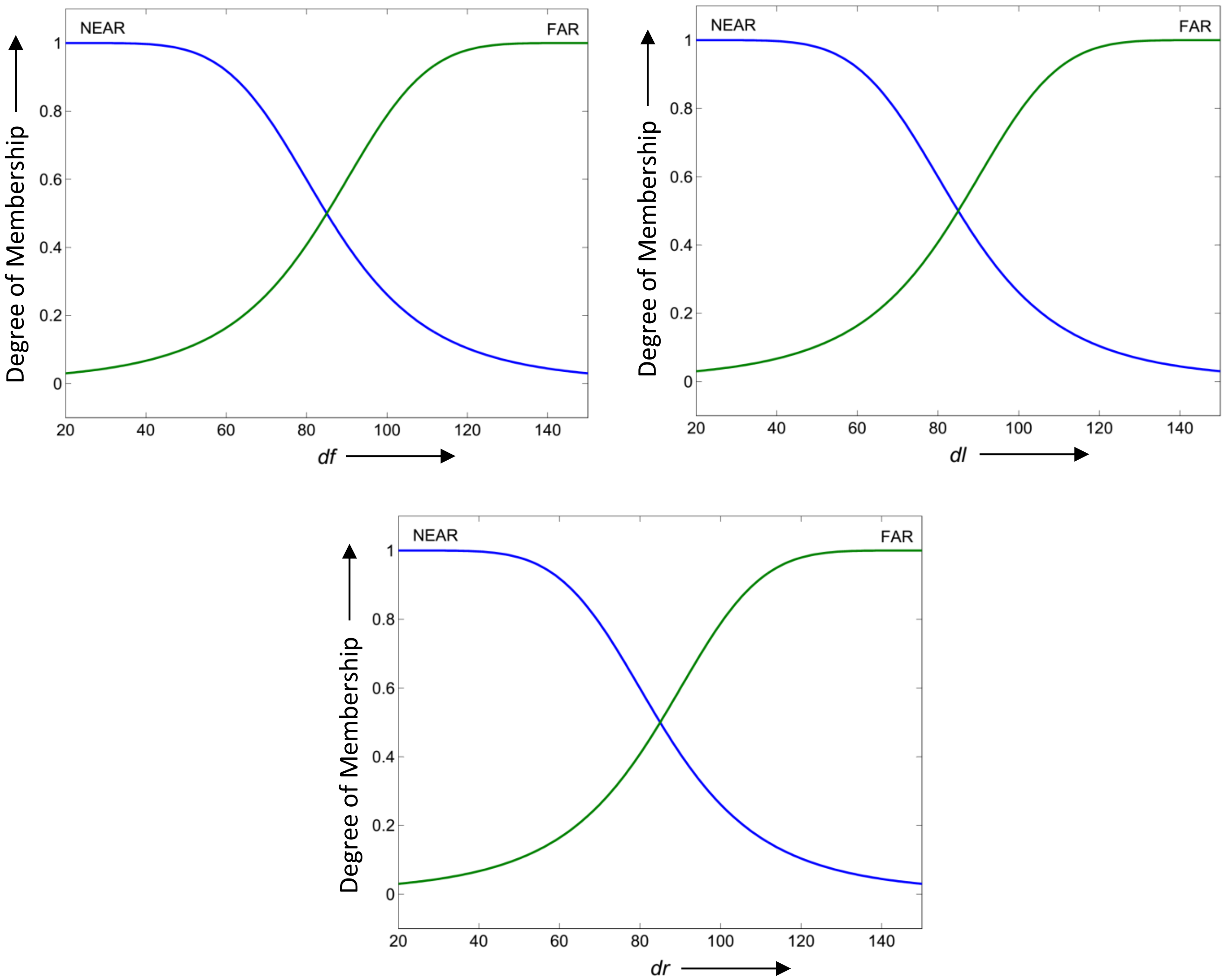


Figure 8.3: Fuzzy membership functions for the inputs (d_f, d_l, and d_r).

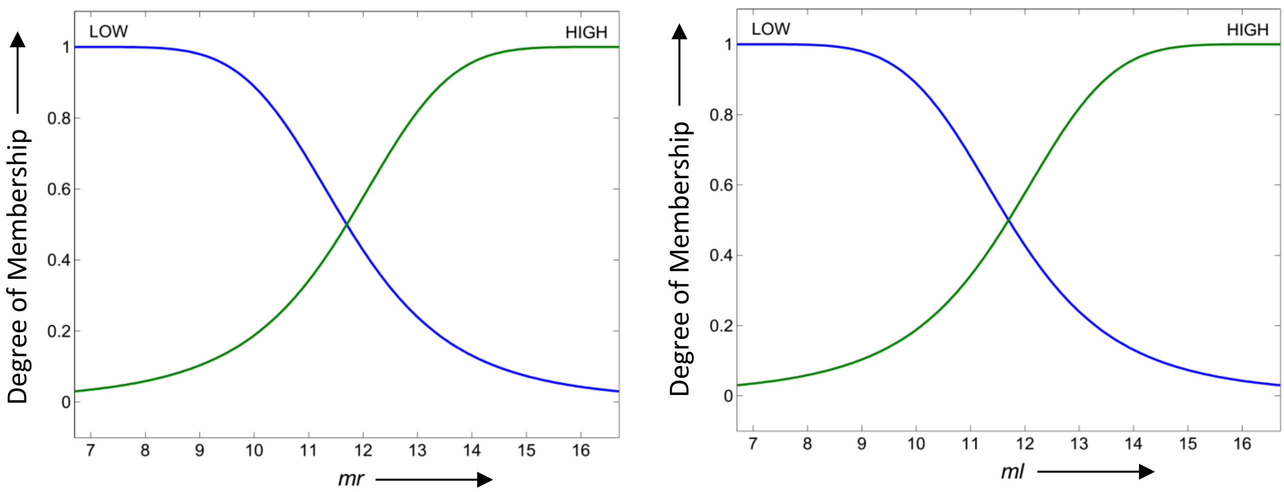


Figure 8.4: Fuzzy membership functions for the outputs (m_r, and m_l).

Table 8.2: Adjusting parameters of the inputs before optimization

Inputs	Membership function	a	b	c
d_f	Near	65	2.5	20
	Far	65	2.5	150
d_l	Near	65	2.5	20
	Far	65	2.5	150
d_r	Near	65	2.5	20
	Far	65	2.5	150

Table 8.3: Adjusting parameters of the outputs before optimization

Outputs	Membership function	a	b	c
m_r	Low	5	2.5	6.7
	High	5	2.5	16.7
m_l	Low	5	2.5	6.7
	High	5	2.5	16.7

8.3 Fuzzy-WDO Algorithm for the Mobile Robot Navigation

This section describes the WDO algorithm used for the membership function parameter optimization of the S-Fuzzy controller for the optimum navigation and collision avoidance in an unknown static and dynamic environment. One major problem with the fuzzy logic is the difficulty of constructing and tuning the correct membership function grade [174]. Because of this problem, the WDO algorithm is used to tune the adjusting parameters of the inputs and outputs. From the Section 8.2, two Gbell membership functions are considered for the inputs (d_f, d_l, and d_r) and outputs (m_r, and m_l). Each Gbell membership function has three adjusting parameters (a, b, and c). Therefore, each input has six adjusting parameters. Similarly, each output has six adjusting parameters. So the total number of adjusting parameters is to be thirty {5 (3 inputs + 2 outputs) × 2 (membership function) × 3 (adjusting parameters a, b, and c) = 30}.

The ranges of adjusting parameters are defined as [$a_{\min}$, $a_{\max}$], [$b_{\min}$, $b_{\max}$] and [$c_{\min}$, $c_{\max}$] respectively, for lower and upper boundary of the WDO algorithm. The $a_{\min}$ and $a_{\max}$ are 30 and 65 for the membership function of the inputs, respectively. The $b_{\min}$ and $b_{\max}$ are 1 and 3.5 for the membership function of the inputs, respectively. The parameters $c_{\min}$ and $c_{\max}$ are 20 and 150 for the membership function of inputs, respectively. Similarly, the $a_{\min}$ and $a_{\max}$ are 2 and 5 for the membership function of outputs, respectively. The $b_{\min}$ and $b_{\max}$ are 1 and 3.5 for the membership function of the outputs, respectively. The parameters $c_{\min}$ and $c_{\max}$ are located at 6.7 and 16.7 for the membership function of outputs, respectively. Figure 8.5 shows the air parcels representation of the WDO algorithm. The optimized membership functions of the inputs (d_f, d_l, and d_r) and the outputs (m_r, and m_l) are shown in Figures 8.6 and 8.7, respectively. The results of the adjusting parameters (a, b, and c) of the inputs and outputs after optimization are listed in Table 8.4 and Table 8.5, respectively.

The most important step in applying the WDO algorithm is to select the fitness function, which is used to evaluate the optimum pressure of the air parcels. In during the optimization process, the combined root mean square errors (CRMSE) are used to evaluate the fitness of the fuzzy controller: -

$$\mathrm{RMSE}_{m_r} = \sqrt{\frac{1}{z}\sum_{p=1}^{z}\left(m_r^{\mathrm{Actual}} - m_r^{\mathrm{FW}}\right)^2} \tag{8.9}$$

$$\mathrm{RMSE}_{m_l} = \sqrt{\frac{1}{z}\sum_{p=1}^{z}\left(m_l^{\mathrm{Actual}} - m_l^{\mathrm{FW}}\right)^2} \tag{8.10}$$

$$\mathrm{CRMSE} = \mathrm{RMSE}_{m_r} + \mathrm{RMSE}_{m_l} \tag{8.11}$$

where m_r^{actual} and m_l^{actual} are the actual value of right and left motor velocity, respectively. The m_r^{FW}, and m_l^{FW} are calculated value of the right and left motor velocity, respectively through Fuzzy-WDO. The RMSE_{m_r}, and RMSE_{m_l} are the root mean square error of the right and left motor velocity respectively; and z is the iteration number.

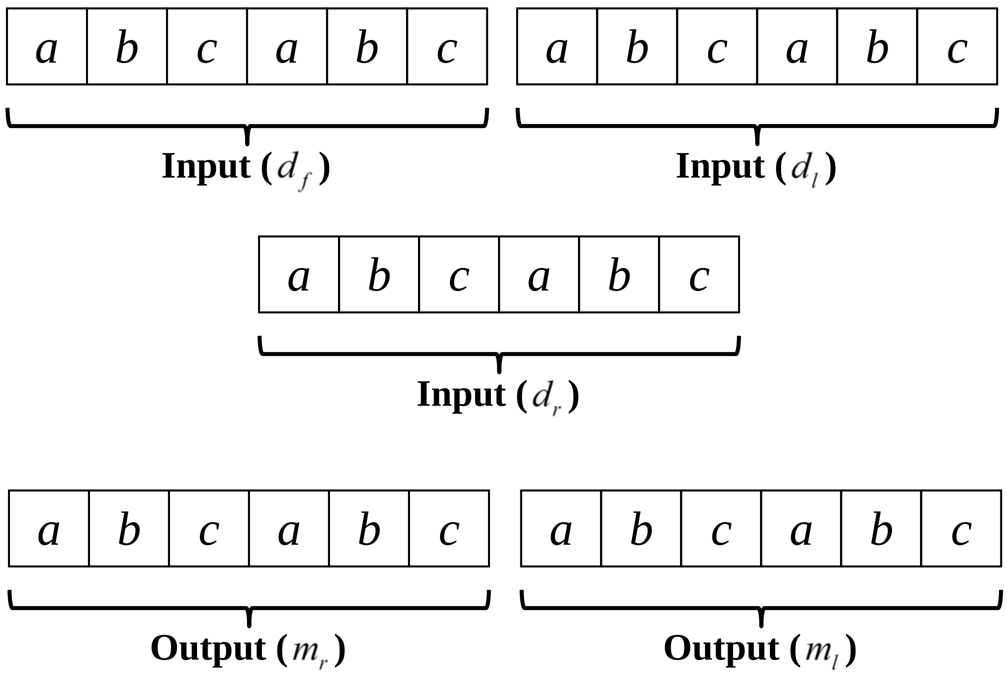


Figure 8.5: Air parcels representation of the WDO algorithm.

Table 8.4: Adjusting parameters of the inputs after optimization

Inputs	**Membership function**	a	b	c
d_f	Near	55.11	2.14	25
	Far	59.6	1.88	149.4
d_l	Near	58.3	2.44	22.4
	Far	62.41	1.76	148.3
d_r	Near	57.42	2.33	23.1
	Far	60.29	1.55	148.9

Table 8.5: Adjusting parameters of the outputs after optimization

Outputs	Membership function	a	b	c
m_r	Low	3.61	2.601	6.515
	High	4.22	2.14	16.2
m_l	Low	3.97	2.21	5.96
	High	4.32	2.96	16.4

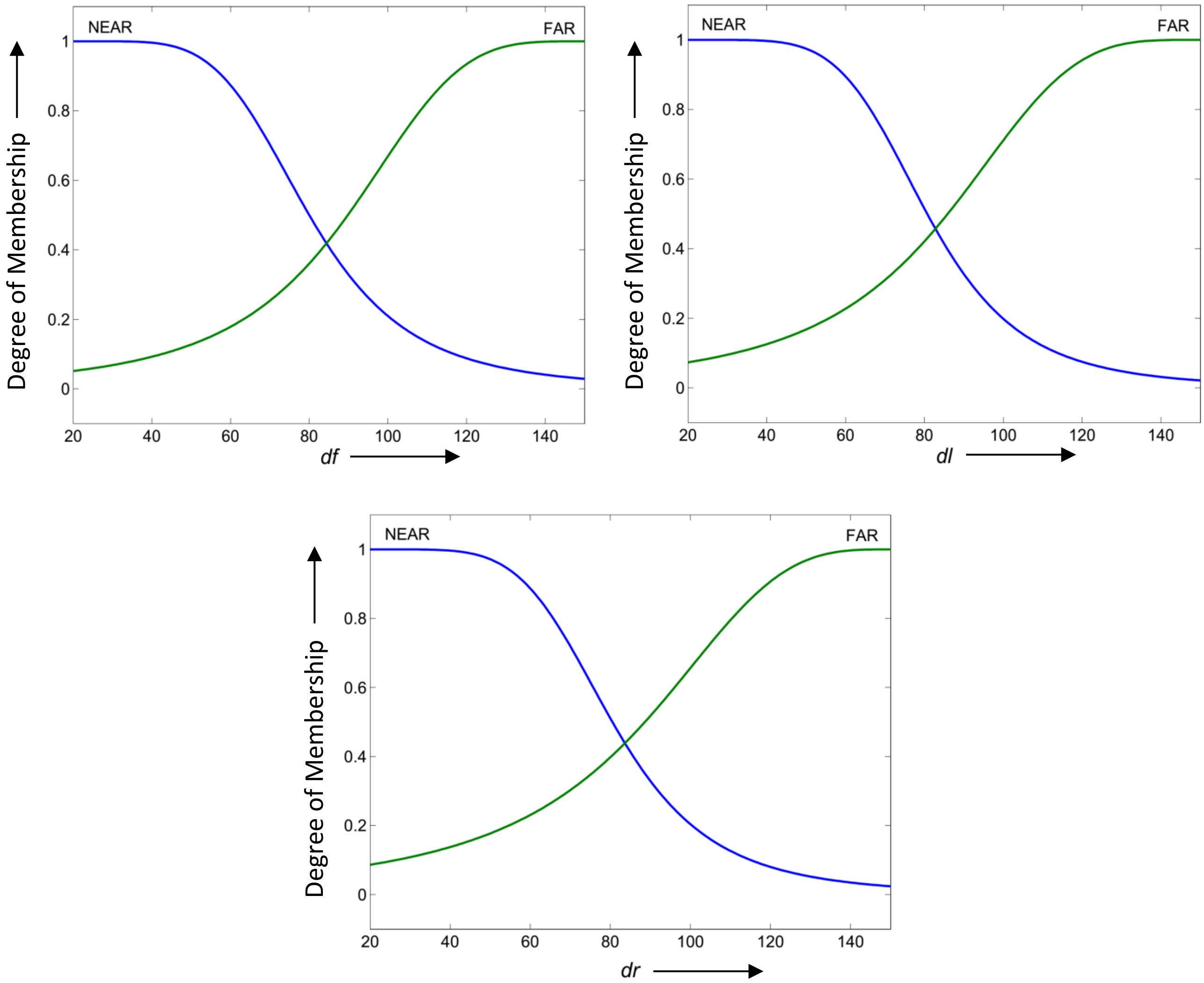


Figure 8.6: Fuzzy membership functions for the inputs (d_f, d_l, and d_r) after optimization.

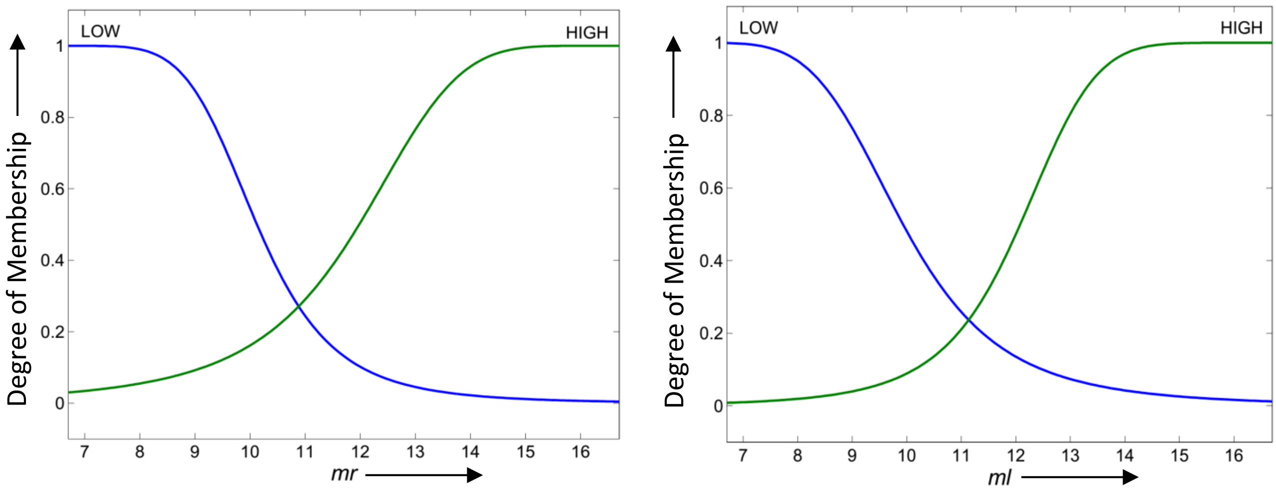
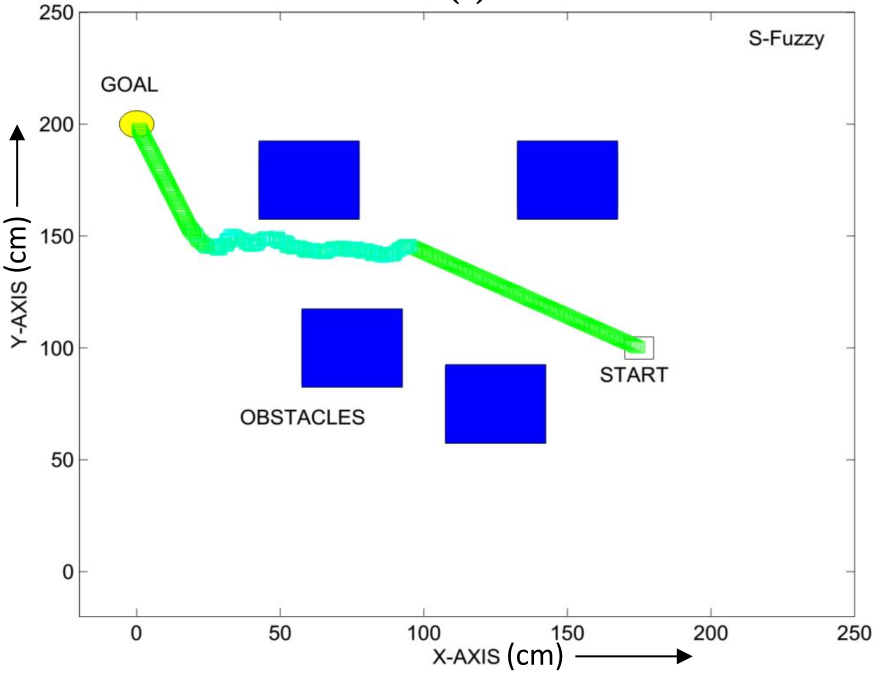


Figure 8.7: Fuzzy membership functions for the outputs (m_r , and m_l) after optimization.

8.4 Simulation Results

This section describes the simulation results using S-Fuzzy and Fuzzy-WDO controllers in the various unknown static and dynamic environments. The simulations are conducted using the MATLAB software on the HP 3.40 GHz processor. Figures 8.8 and 8.9 show the navigation result of the mobile robot between the obstacles and walls respectively, using the S-Fuzzy and Fuzzy-WDO controller in the unknown environments. Similarly, the Figure 8.10 demonstrates the navigation of a mobile robot in an unknown environment with the presence of two dynamic obstacles using Fuzzy-WDO controller. It is assumed that the position of the start point and goal point are known. But the positions of all the obstacles in the environment are unknown for the robot. In the simulation results, the green and red color trajectory indicates the path generated by the S-Fuzzy and Fuzzy-WDO controllers, respectively. Simulation results show the Fuzzy-WDO controller gives smooth and optimal path compared to the S-Fuzzy controller. Table 8.6 shows the navigation path length and time taken by the robot using the S-Fuzzy and Fuzzy-WDO controller in the various unknown environments.

(a)



(b)

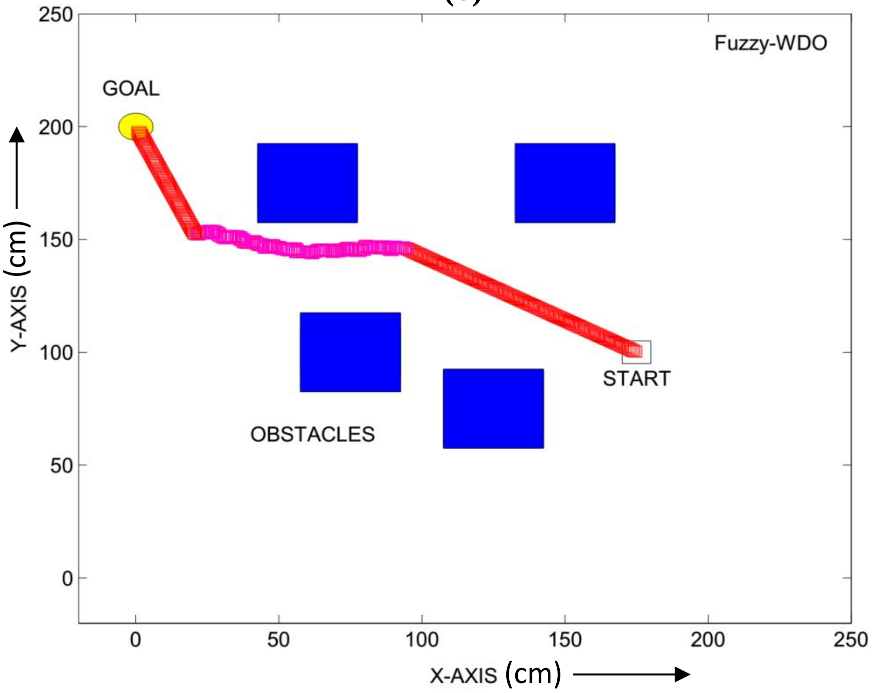


Figure 8.8: Mobile robot navigation between the obstacles using **(a)** S-Fuzzy and **(b)** Fuzzy-WDO controller.

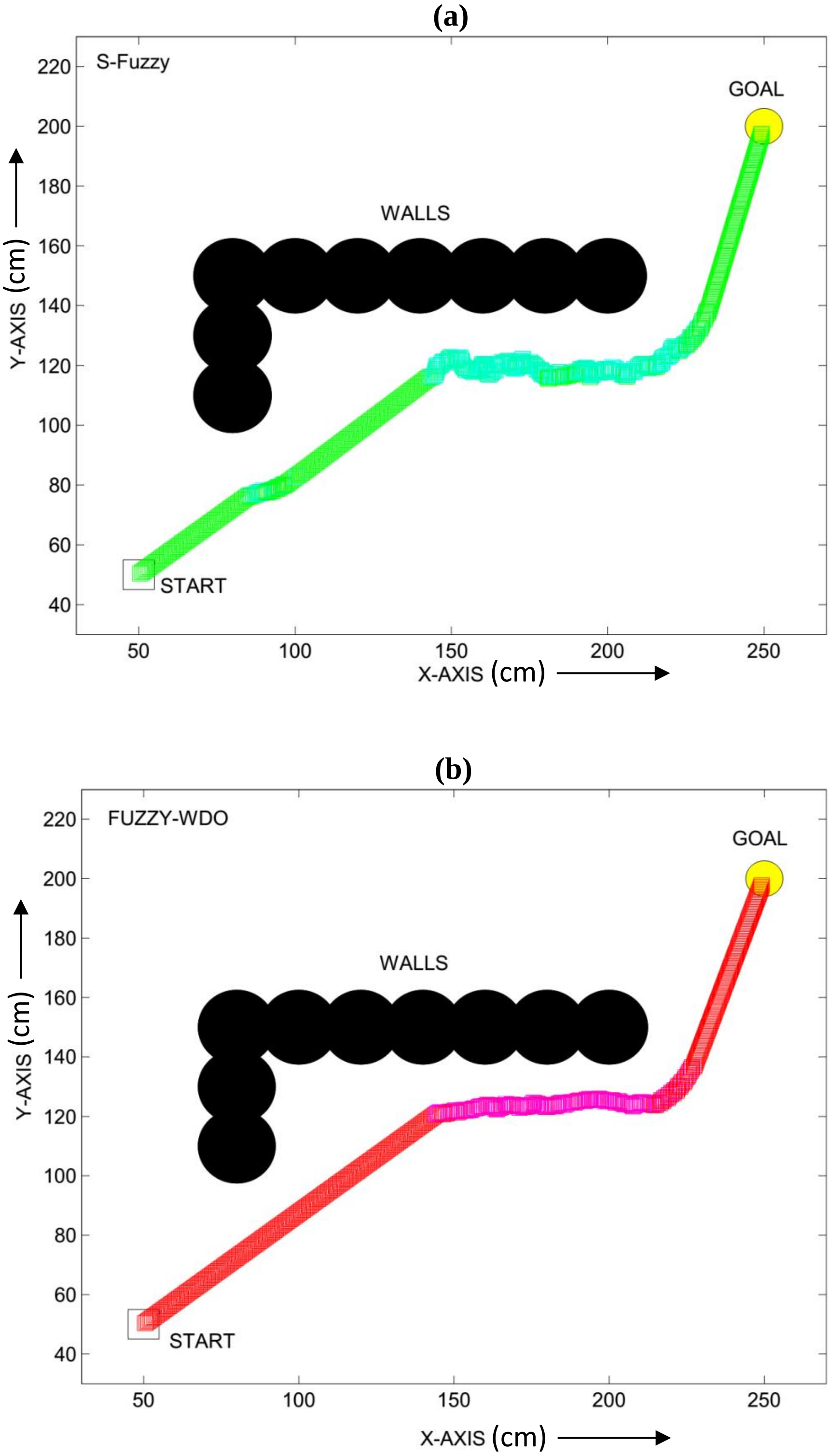


Figure 8.9: Mobile robot navigation between the walls using **(a)** S-Fuzzy and **(b)** Fuzzy-WDO controller.

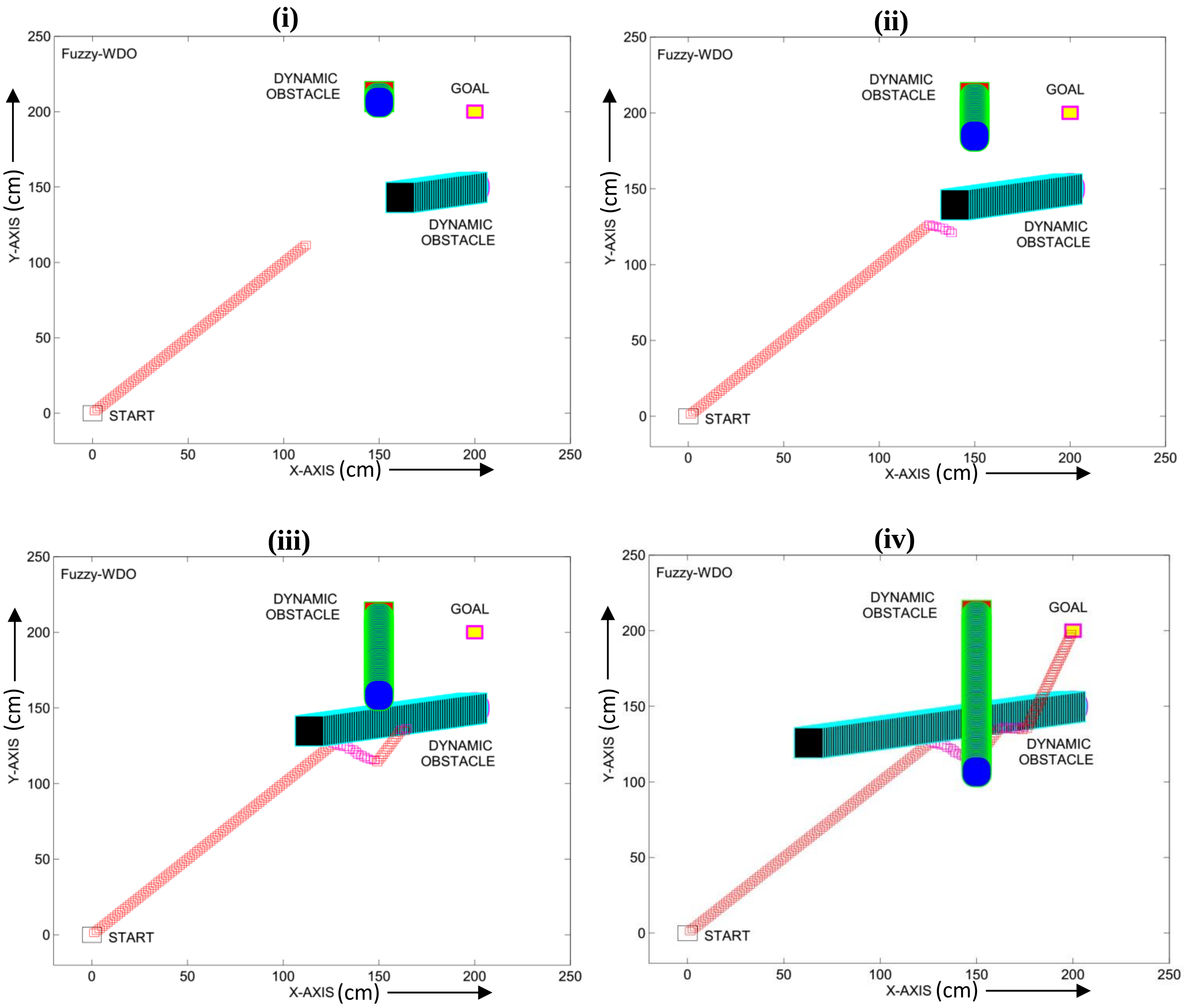


Figure 8.10: Mobile robot navigation in the dynamic environment using Fuzzy-WDO controller.

Table 8.6: The simulation results of S-Fuzzy and Fuzzy-WDO controllers

Figure no.	Controller	Navigation path length (cm)	Navigation time (sec)
Figure 8.8	S-Fuzzy	79	7.2
	Fuzzy-WDO	74	6.9
Figure 8.9	S-Fuzzy	104	9.1
	Fuzzy-WDO	98	8.7

8.5 Comparison with Previous Works

This section describes the computer simulation result comparison between the previous model [45] and proposed Fuzzy-WDO controller in the same path planning problems. In the article [45], the authors have used two simple fuzzy controllers such as tracking fuzzy logic control (TFLC) and obstacle avoidance fuzzy logic control (OAFLC) without adjusting its membership function for mobile robot navigation. Figures 8.11 and 8.12 show the mobile robot navigation in the same environment without obstacle using fuzzy controller [45] and proposed Fuzzy-WDO controller, respectively. Similarly, Figures 8.13 and 8.14 present the path covered by the robot in the same environment with the four obstacles using fuzzy controllers [45] and proposed Fuzzy-WDO controller, respectively. From the simulation figures, it can be seen that the proposed Fuzzy-WDO controller covers shorter distance to reach the goal as compared to previous model [45] because WDO algorithm adjusts the membership function of the fuzzy controller, which provides better result compared to the standalone fuzzy model. Table 8.7 illustrates the path traced (in cm) by the robot to reach the goal using proposed controller and previous model [45]. The centimeter measurements are taken on the proportional basis.

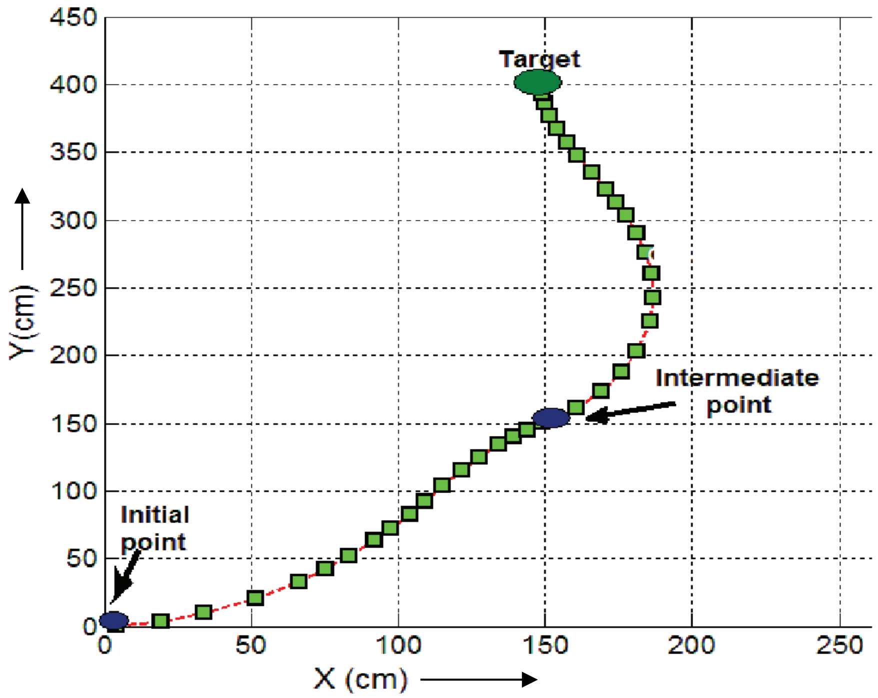


Figure 8.11: Mobile robot navigation in an environment without obstacle using fuzzy controller [45].

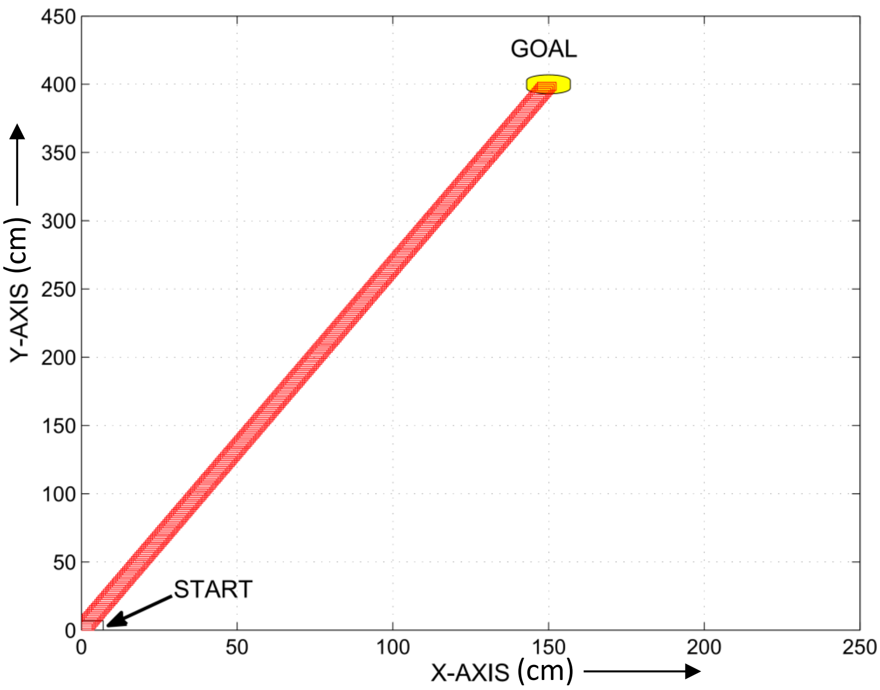


Figure 8.12: Mobile robot navigation in an environment without obstacle using Fuzzy-WDO controller.

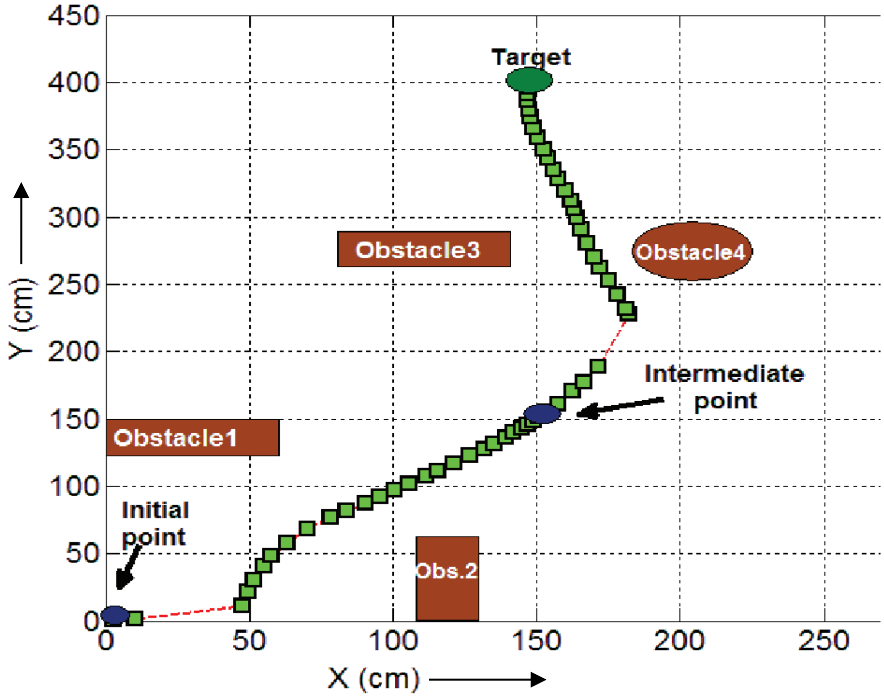


Figure 8.13: Mobile robot navigation in an environment with four obstacles using fuzzy controller [45].

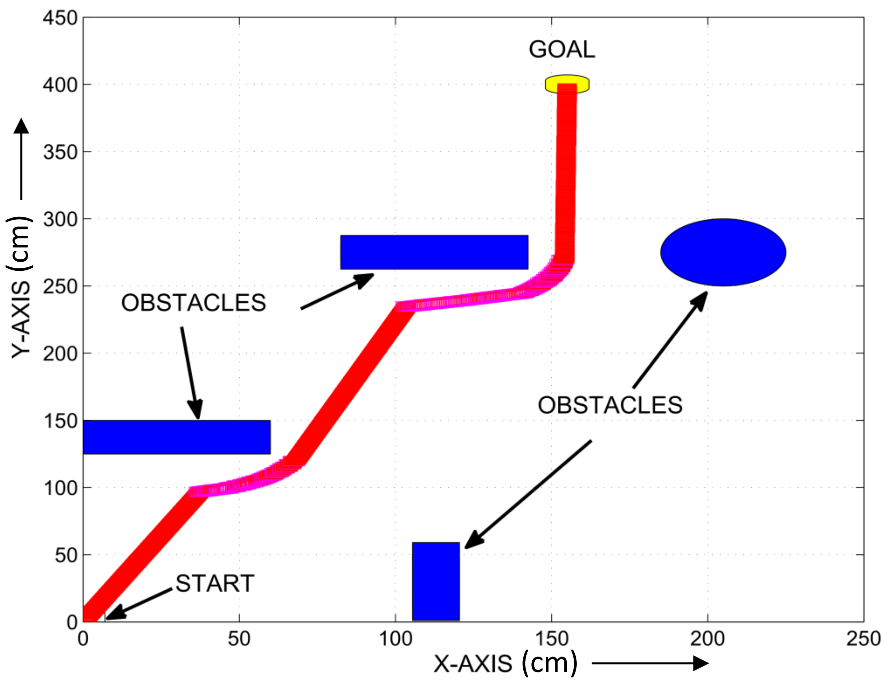


Figure 8.14: Mobile robot navigation in an environment with four obstacles using Fuzzy-WDO controller.

Table 8.7: The simulation result comparison between the fuzzy controller [45] and proposed Fuzzy-WDO controller

Figure no.	**Method**	**Navigation path length (cm)**
Figure 8.11	Fuzzy controller [45]	181
Figure 8.12	Fuzzy-WDO controller	165
Figure 8.13	Fuzzy controller [45]	183
Figure 8.14	Fuzzy-WDO controller	173

8.6 Experimental Results

8.6.1 Khepera-III Mobile Robot Description

The experiments are conducted using the Khepera-III mobile robot in unknown environments. The Khepera-III mobile robot has two wheels controlled by two DC servo motors and one caster wheel. The diameter and height of the robot are 13cm and 7cm respectively. The Khepera-III mobile robot is equipped with nine infrared proximity sensors and five ultrasonic sensors, as shown in Figure 8.15. The infrared proximity sensor reads obstacles up to 30cm, and the ultrasonic sensor reads obstacles from 20cm to 4m approximately. In this study, we have set the minimum and maximum velocity of Khepera-III mobile robot between the 6.7-16.7cm/sec.

8.6.2 Experiments

In the experiments, the controllers are implemented in the Khepera-III mobile robot using HP laptop. The width and height of the experimental platform are 250cm and 250cm, respectively. Figures 8.16 and 8.17 show the real time navigation of the Khepera-III mobile robot in unknown environments with the obstacles and walls, respectively. In Figure 8.16, the start position of the robot is (175, 100) cm and the position of the goal is (0, 200) cm. The starting angle between the robot and the goal is 29.74°. In Figure 8.17, the start position of the robot is (50, 50) cm and the goal position is (250, 200) cm. The starting angle between the robot and the goal is 36.87°. In the experiments, it is assumed that the position of the start point and goal point are known, but the positions of all the obstacles in the environment are unknown for the robot. The S-Fuzzy and Fuzzy-WDO controller generates the motor velocity control commands of the robot for obstacle avoidance using on-board sensor information. The successful experimental results in the various unknown environments are shown below to verify the effectiveness of the S-Fuzzy and Fuzzy-WDO controllers. Table 8.8 shows the experimental path length and time taken by the Khepera-III mobile robot to reach target using the S-Fuzzy and Fuzzy-WDO controllers in the various unknown environments. Tables 8.9 and 8.10 present the travelling path length and navigation time comparison between the simulation and experimental results, respectively. In the comparison study between the simulation and experiments, it is observed that errors within 6% have been found, these are happened due to slippage and friction during real time experiment.

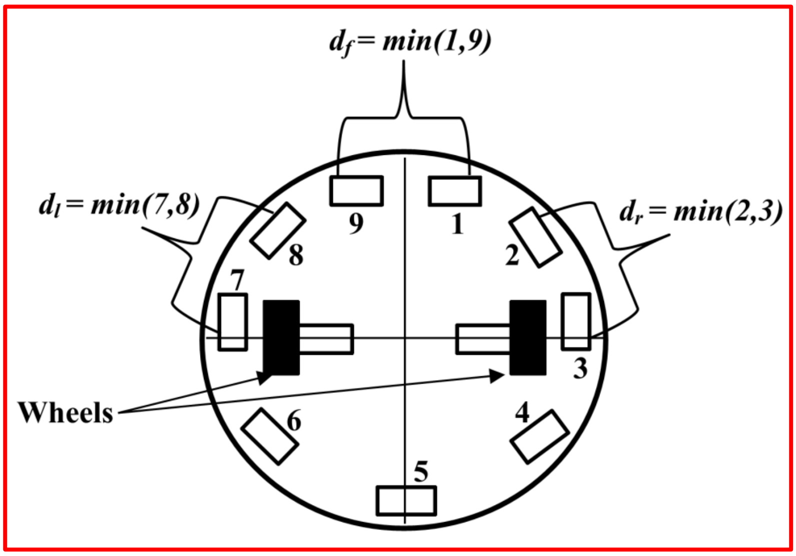


Figure 8.15: Infrared proximity sensor distribution of Khepera-III mobile robot.

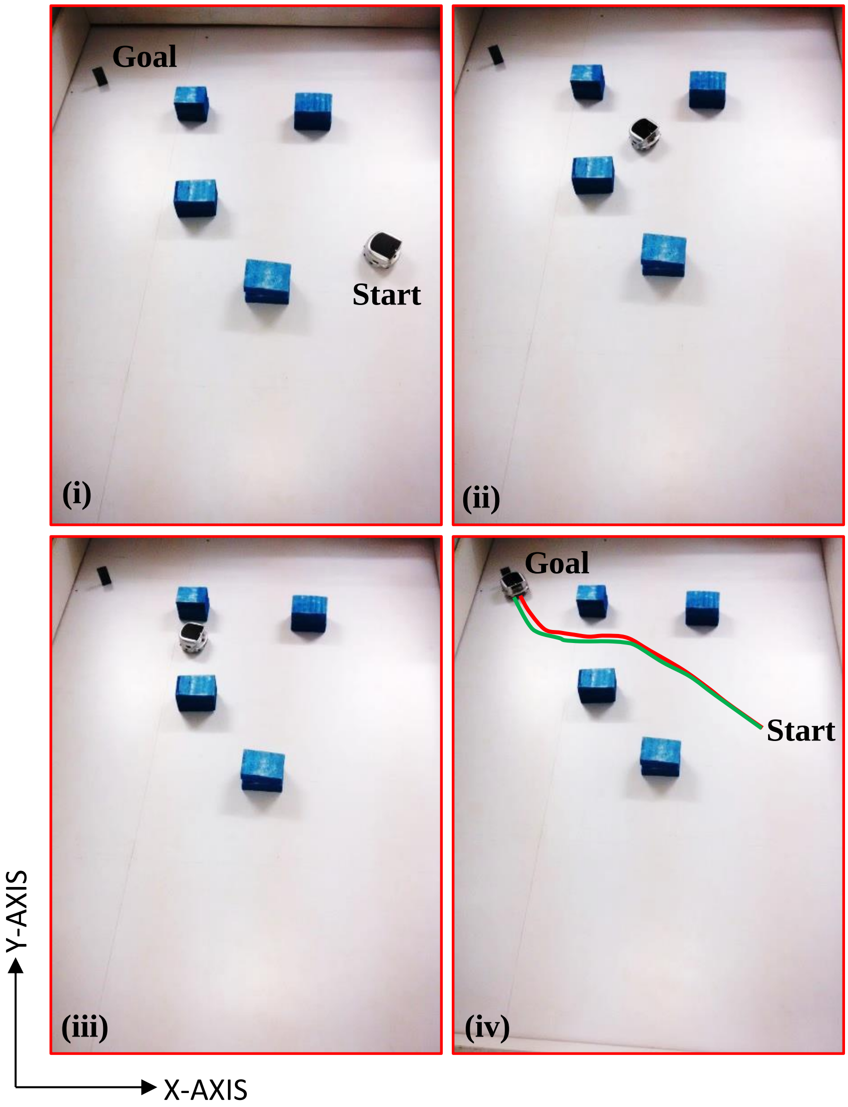


Figure 8.16: Real time navigation of Khepera-III mobile robot between the obstacles using S-Fuzzy and Fuzzy-WDO controller.

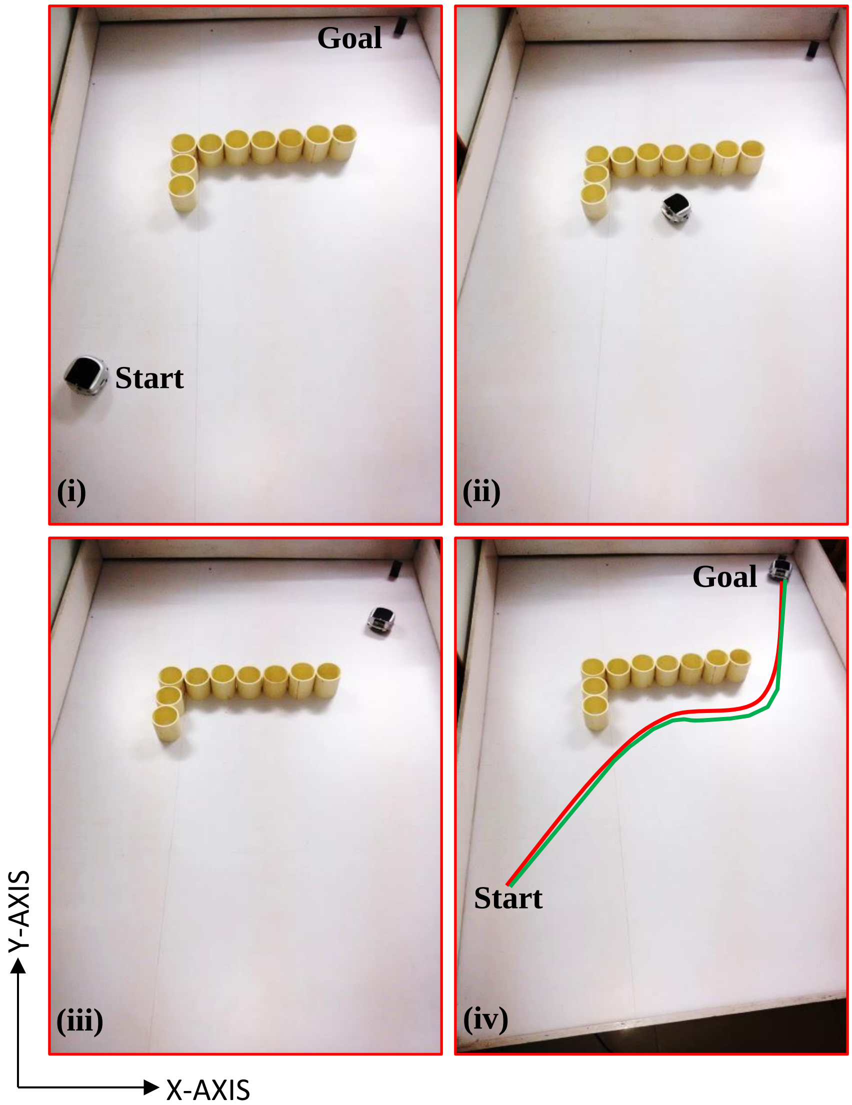


Figure 8.17: Real time navigation of Khepera-III mobile robot between the walls using S-Fuzzy and Fuzzy-WDO controller.

Table 8.8: The experimental results of S-Fuzzy and Fuzzy-WDO controllers

Figure no.	**Controller**	**Experimental path length (cm)**	**Experimental time (sec)**
Figure 8.16	S-Fuzzy	84	7.7
	Fuzzy-WDO	78	7.3
Figure 8.17	S-Fuzzy	111	9.7
	Fuzzy-WDO	103	9.2

Table 8.9: Travelling path lengths comparison between simulation and experimental results

Figure no. (Simulation and experimental res.)	**Controller**	**Travelling path length (cm)**		**Error between simulation and experimental results**
		Simulation result	**Experimental result**	
Figure 8.8 and 8.16	S-Fuzzy	79	84	5.95%
	Fuzzy-WDO	74	78	5.12%
Figure 8.9 and 8.17	S-Fuzzy	104	111	6.31%
	Fuzzy-WDO	98	103	4.85%

Table 8.10: Navigation time comparison between simulation and experimental results

Figure no. (Simulation and experimental res.)	**Controller**	**Navigation time (sec)**		**Error between simulation and experimental results**
		Simulation result	**Experimental result**	
Figure 8.8 and 8.16	S-Fuzzy	7.2	7.7	6.49%
	Fuzzy-WDO	6.9	7.3	5.47%
Figure 8.9 and 8.17	S-Fuzzy	9.1	9.7	6.18%
	Fuzzy-WDO	8.7	9.2	5.43%

8.7 Summary

In this chapter, the two methods S-Fuzzy controller and the hybrid Fuzzy-WDO algorithm have been applied to the mobile robot navigation. A new population-based optimization algorithm, called Wind Driven Optimization (WDO) is used to optimize and set the antecedent and consequent parameters of the fuzzy controller. The proposed algorithms are successfully verified through simulations and real-time experiments in the different environments. Simulation and experimental results demonstrate that the Fuzzy-WDO controller provide better performance as compared to the S-Fuzzy controller. In the comparison study between the simulation and experiment results errors are recorded, and the errors are found due to the effect of slippage and friction between the wheels of the robot and surface during navigation in real time mode. During experiment utmost care has been taken to minimize the slippage and friction between the wheels and surface. Still the effect of slippage and friction are unavoidable, and errors are recorded during the comparison of the results for travelling path length (5.56%) and for navigation time (5.9%).

Chapter 9

Comparative Study of the Proposed Soft Computing Techniques Applied for Mobile Robot Navigation

9.1 Introduction

In the previous chapters, the various soft computing techniques such as H-Fuzzy architecture, CN-Fuzzy architecture, Fuzzy-SA algorithm, WDO algorithm, and Fuzzy-WDO algorithm have been successfully designed and implemented for the mobile robot navigation and obstacle avoidance in the different static and dynamic environments. This chapter presents the comparative analysis of the all developed techniques in the two different simulation and experimental test environments. In the comparative study, it is found that the Fuzzy-WDO algorithm is more efficient (in terms of path length and navigation time) as compared to rest of the techniques for mobile robot navigation. Rest of the chapter is organized as follows: Section 9.2 presents the simulation result comparison of the all developed soft computing techniques. Section 9.3 demonstrates the experimental verification of the developed simulations. Finally, Section 9.4 depicts the summary.

9.2 Simulation studies

All the simulations are performed through MATLAB software in the HP 3.40 GHz processor personal computer. The simulation studies are divided into two test environments: test1 and test2, respectively, which are presented below.

9.2.1 Simulation Test1

In the first simulation test, all the developed soft computing techniques are employed for mobile robot navigation and obstacle avoidance in the environment. The dimensions of the environment are 250cm width and 250cm height. Figures 9.1 (a) to 9.1 (e) show the best results of the mobile robot navigation between the obstacles using H-Fuzzy architecture, CN-Fuzzy architecture, Fuzzy-SA algorithm, WDO algorithm, and Fuzzy-WDO algorithm, respectively. The start position of the mobile robot is (175, 100) cm and the position of the goal is (0, 200) cm.

9.2.2 Simulation Test2

In the second simulation test, the mobile robot navigates between the walls in the environment using all developed soft computing techniques. The width and height of the environment are 270cm and 240cm, respectively. Figures 9.2 (a) to 9.2 (e) illustrate the best results of the mobile robot navigation in the test environment using H-Fuzzy architecture, CN-Fuzzy architecture, Fuzzy-SA algorithm, WDO algorithm, and Fuzzy-WDO algorithm, respectively. The start and goal position of the mobile robot are (50, 50) cm and (250, 200) cm, respectively.

In both the simulation tests (test1 and test2), it is assumed that the position of the start point and goal point are known, but the positions of the obstacles and walls are unknown for the robot. A minimum threshold distance has been fixed between the robot and the obstacles. If the robot detects the obstacles in the threshold range, then the proposed techniques provide the motion and orientation control command to the robot for avoiding the obstacle and wall.

From both the simulation tests, it is observed that the Fuzzy-WDO algorithm covers shorter distance to reach the goal as compared to the rest of the techniques because WDO algorithm adjusts the membership function of the fuzzy controller, which provides better result compared to the other techniques. For each simulation test, we have run all the developed techniques for 10-15 times, and the average path length and time have been listed in the table. Table 9.1 shows the average navigation path length (in cm) and time taken (in the sec) by the robot to reach the goal using all the developed techniques in different environments.

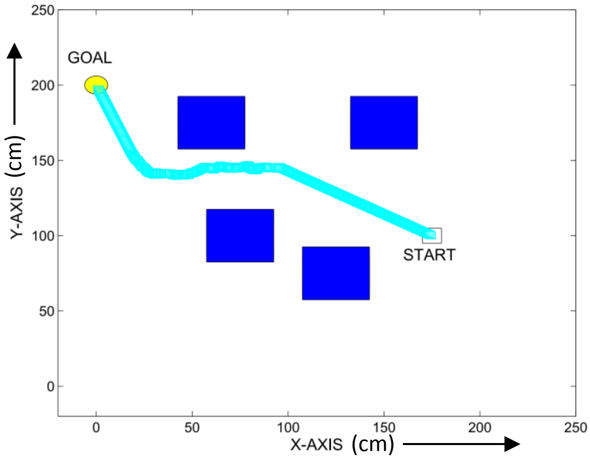


Figure 9.1 **(a)**:

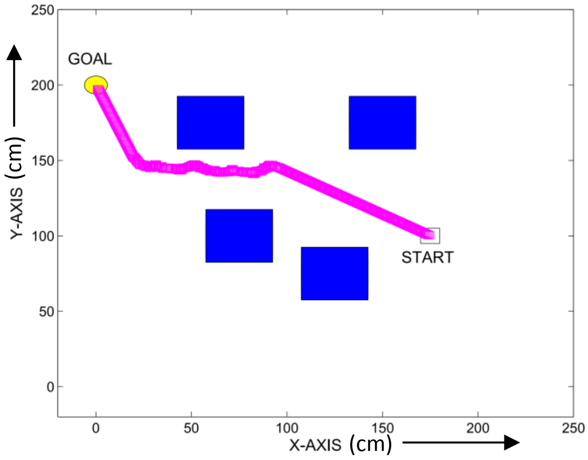


Figure 9.1 **(b)**:

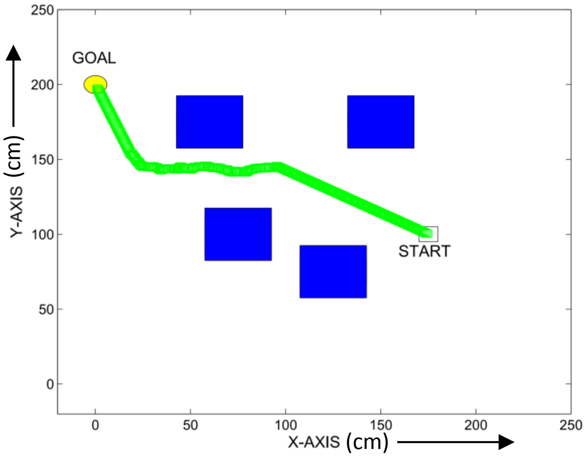


Figure 9.1 **(c)**:

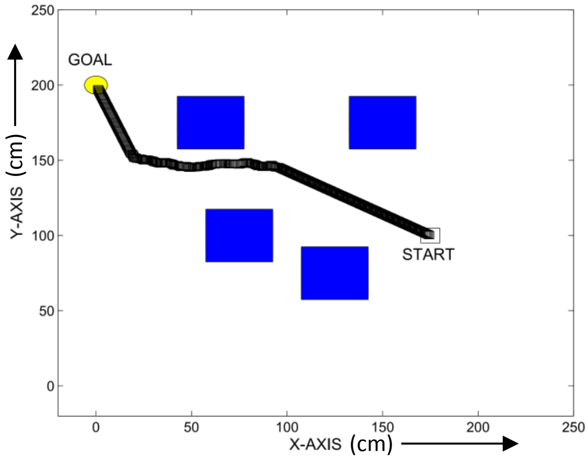


Figure 9.1 **(d)**:

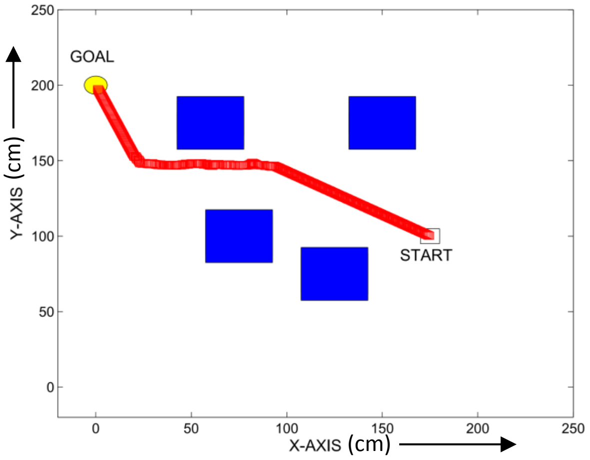


Figure 9.1 **(e)**:

Figure 9.1 **(a)**: H-Fuzzy architecture.

Figure 9.1 **(b)**: CN-Fuzzy architecture.

Figure 9.1 **(c)**: Fuzzy-SA algorithm.

Figure 9.1 **(d)**: WDO algorithm.

Figure 9.1 **(e)**: Fuzzy-WDO algorithm.

Figure 9.1: Mobile robot navigation and obstacle avoidance in the simulation test1 using the developed soft computing techniques.

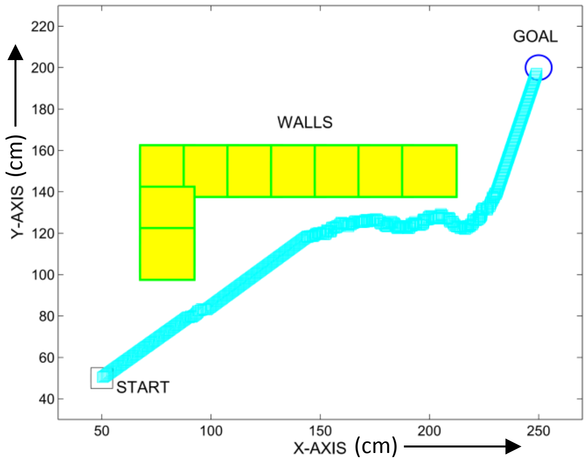


Figure 9.2 **(a)**:

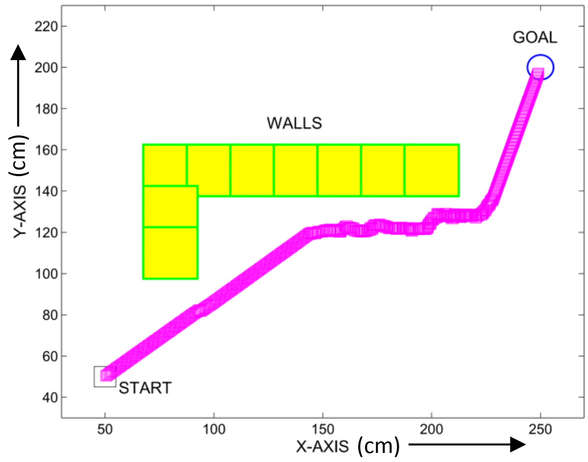


Figure 9.2 **(b)**:

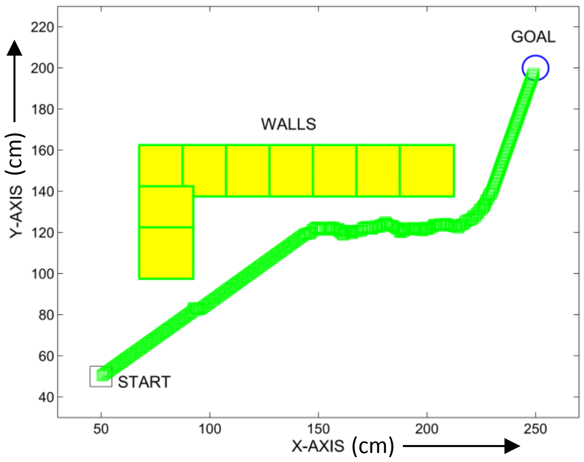


Figure 9.2 **(c)**:

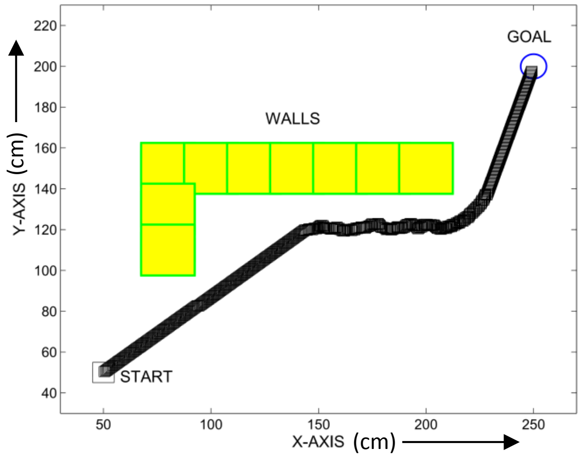


Figure 9.2 **(d)**:

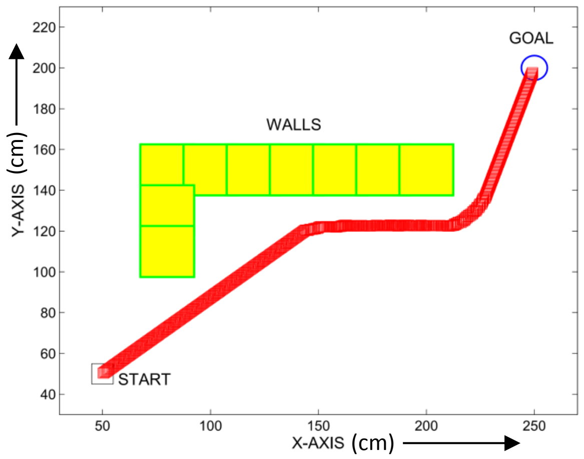


Figure 9.2 **(e)**:

Figure 9.2 **(a)**: H-Fuzzy architecture.

Figure 9.2 **(b)**: CN-Fuzzy architecture.

Figure 9.2 **(c)**: Fuzzy-SA algorithm.

Figure 9.2 **(d)**: WDO algorithm.

Figure 9.2 **(e)**: Fuzzy-WDO algorithm.

Figure 9.2: Mobile robot navigation and obstacle avoidance in the simulation test2 using the developed soft computing techniques.

Table 9.1: Simulation results of the mobile robot navigation in the test1 and test2 using all developed techniques

Figure no.	Techniques	Navigation path length (cm)	Navigation time (sec)
Figure 9.1 (a) to (e) (Test1)	H-Fuzzy	80	7.7
	CN-Fuzzy	78	7.5
	Fuzzy-SA algorithm	76	7.4
	WDO algorithm	75	7.2
	Fuzzy-WDO algorithm	**74**	**6.9**
Figure 9.2 (a) to (e) (Test2)	H-Fuzzy	105	9.4
	CN-Fuzzy	102	9.3
	Fuzzy-SA algorithm	101	9.2
	WDO algorithm	100	8.9
	Fuzzy-WDO algorithm	**98**	**8.7**

Note: Bold value indicates the best results compared to rest of the techniques.

9.3 Experimental studies of the developed simulations

This section demonstrates the experimental verification of the developed simulations in test1 and test2. All the experiments have been conducted through C/C++ running Arduino microcontroller based wheeled mobile robot. The other descriptions of the experimental mobile robot have been discussed in the previous chapters and appendix. The size of the experimental platform is 300cm width and 300cm height. The experimental studies are also divided into two test environments: test1 and test 2, respectively, which is presented below.

9.3.1 Experimental Test1

In the experimental test1, all the developed soft computing techniques have experimented between the four square shaped obstacles. The robot moves from the start point (175cm, 100cm) to goal point (0cm, 200cm). The starting angle between the robot and the goal is 29.74°. Figures 9.3 (a) to 9.3 (e) present the experimental verification of the above simulation results (see the Figures 9.1) using H-Fuzzy architecture, CN-Fuzzy architecture, Fuzzy-SA algorithm, WDO algorithm, and Fuzzy-WDO algorithm, respectively. From the experimental results, it can be observed that the robot successfully avoids all the obstacles and reach the goal similar to simulation test1.

9.3.2 Experimental Test2

In the experimental test2, the mobile robot navigates from the start point (50cm, 50cm) to goal point (250cm, 200cm) between the walls in the environment using all developed soft computing techniques. The starting angle between the robot and the goal is 36.87°. Figures 9.4 (a) to 9.4 (e) show the real-time navigation results of the wheeled mobile robot using H-Fuzzy architecture, CN-Fuzzy architecture, Fuzzy-SA algorithm, WDO algorithm, and Fuzzy-WDO algorithm, respectively. In the experimental test2, it can be observed that the trajectory of mobile robot navigation is similar to the trajectory of simulation test2.

In the experimental studies (test1 and test2), we have set a threshold distance (30cm) between the robot and the obstacles. If the robot detects the obstacles in the threshold range, then the developed techniques send the motion and orientation control command to the robot for avoiding the obstacle.

For each experimental test, we have run all the developed techniques for 10-15 times, and the average path length and time have been listed in the table. Table 9.2 illustrates the real-time average navigation path length (in cm) and time taken (in the sec) by the robot to reach the goal using developed techniques in the different environments. From the experimental studies, it can be concluded that the Fuzzy-WDO algorithm covers shorter distance to reach the goal as compared to other developed techniques.

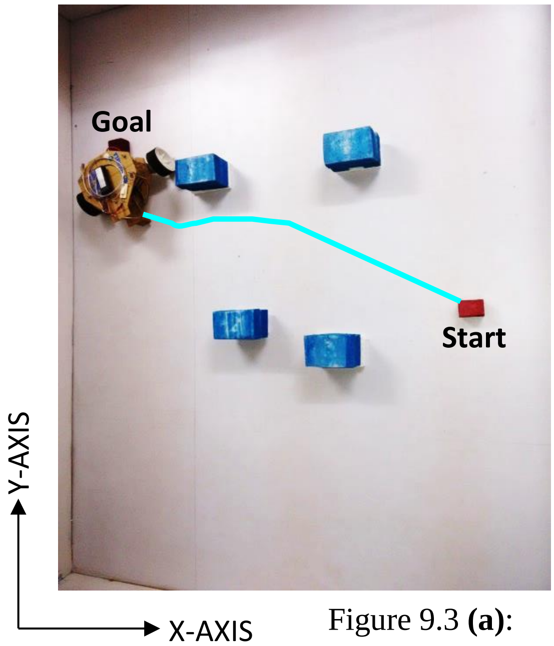


Figure 9.3 **(a)**:

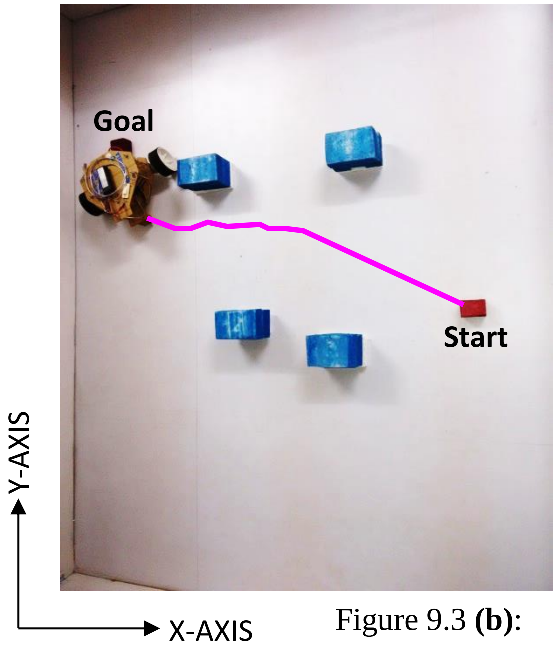


Figure 9.3 **(b)**:

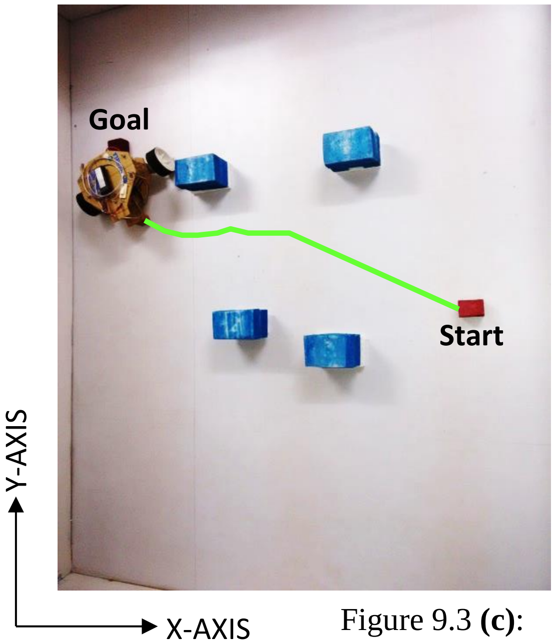


Figure 9.3 **(c)**:

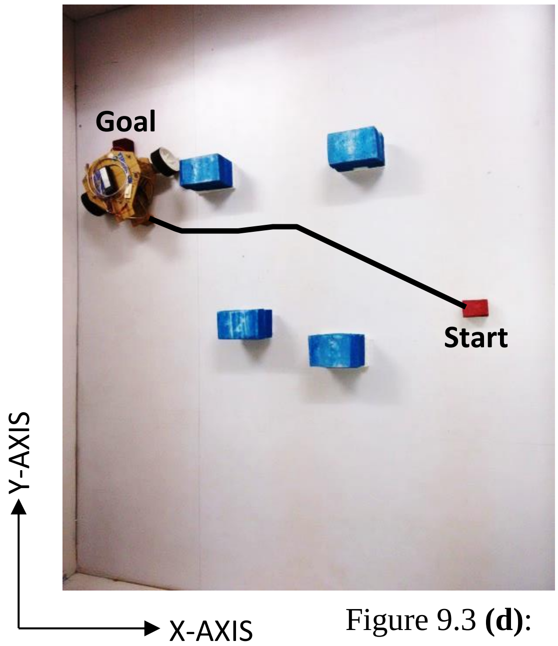


Figure 9.3 **(d)**:

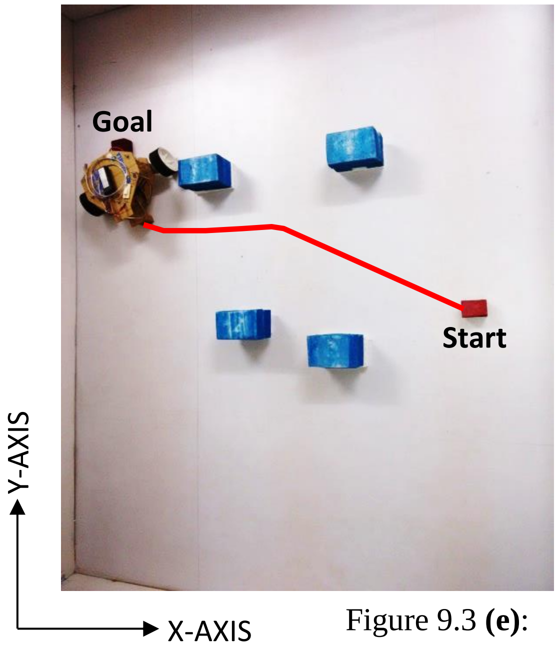


Figure 9.3 **(e)**:

Figure 9.3 **(a)**: H-Fuzzy architecture.

Figure 9.3 **(b)**: CN-Fuzzy architecture.

Figure 9.3 **(c)**: Fuzzy-SA algorithm.

Figure 9.3 **(d)**: WDO algorithm.

Figure 9.3 **(e)**: Fuzzy-WDO algorithm.

Figure 9.3: Mobile robot navigation and obstacle avoidance in the experimental test1 using the developed soft computing techniques.

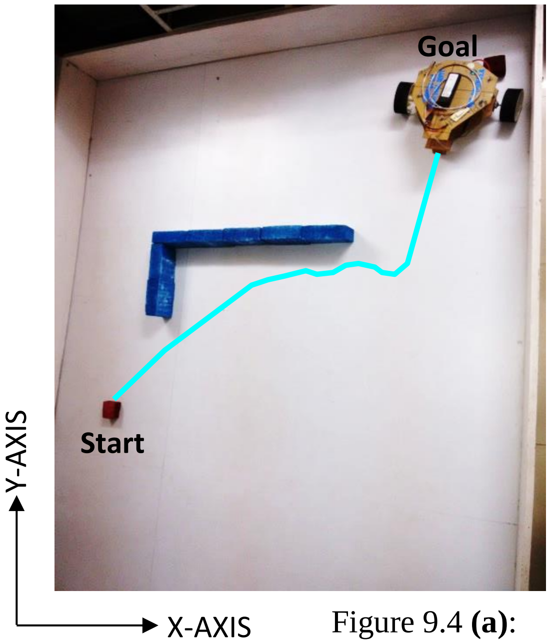


Figure 9.4 **(a)**:

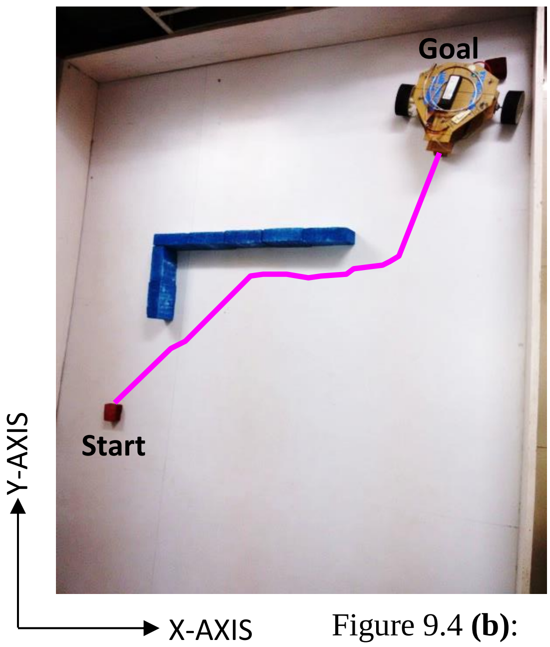
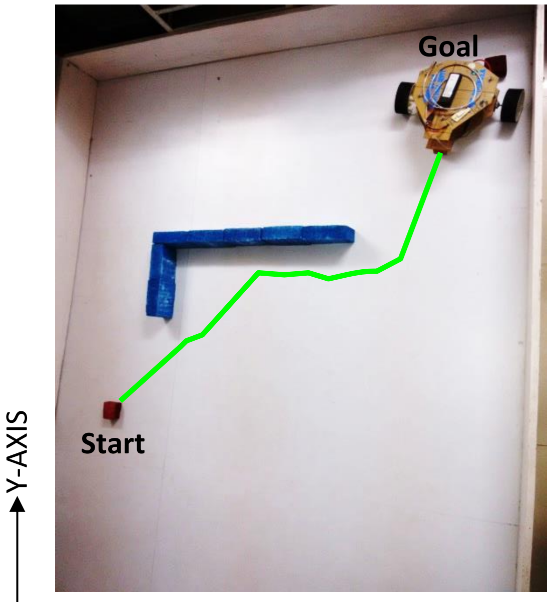
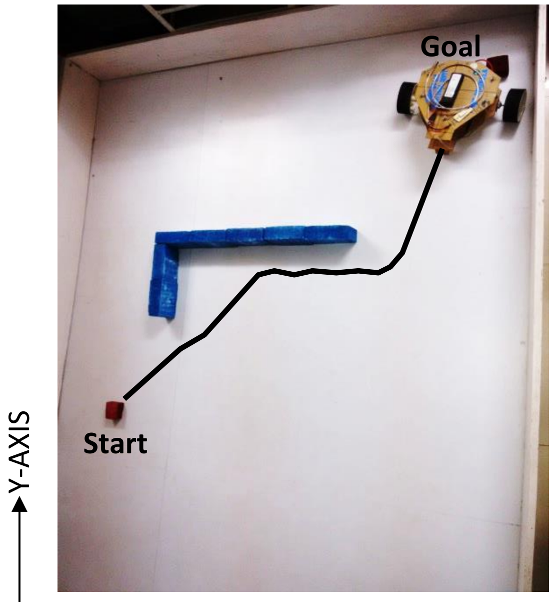


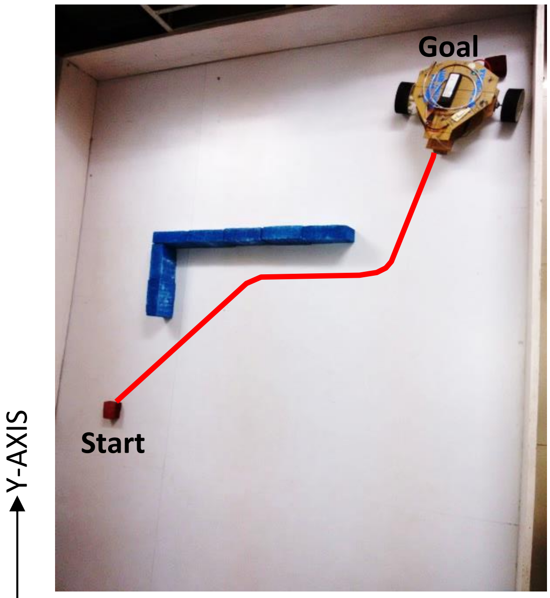
Figure 9.4 **(b)**:



X-AXIS Figure 9.4 **(c)**:



X-AXIS Figure 9.4 **(d)**:



X-AXIS Figure 9.4 **(e)**:

Figure 9.4 **(a)**: H-Fuzzy architecture.

Figure 9.4 **(b)**: CN-Fuzzy architecture.

Figure 9.4 **(c)**: Fuzzy-SA algorithm.

Figure 9.4 **(d)**: WDO algorithm.

Figure 9.4 **(e)**: Fuzzy-WDO algorithm.

Figure 9.4: Mobile robot navigation and obstacle avoidance in the experimental test2 using the developed soft computing techniques.

Table 9.2: Experimental results of the mobile robot navigation in the test1 and test2 using all developed techniques

Figure no.	**Techniques**	**Navigation path length (cm)**		**Navigation time (sec)**	
		Simulation	**Experimental**	**Simulation**	**Experimental**
Figure 9.3 (a) to (e) (Test1)	H-Fuzzy	80	85	7.7	8.2
	CN-Fuzzy	78	83	7.5	8.0
	Fuzzy-SA algorithm	76	81	7.4	7.8
	WDO algorithm	75	80	7.2	7.6
	Fuzzy-WDO algorithm	**74**	**78**	**6.9**	**7.2**
Figure 9.4 (a) to (e) (Test2)	H-Fuzzy	105	112	9.4	10
	CN-Fuzzy	102	109	9.3	9.9
	Fuzzy-SA algorithm	101	107	9.2	9.8
	WDO algorithm	100	105	8.9	9.4
	Fuzzy-WDO algorithm	**98**	**103**	**8.7**	**9.1**

Note: Bold value indicates the best results compared to rest of the techniques.

9.4 Summary

In the present study, all the developed soft computing techniques such as H-Fuzzy architecture, CN-Fuzzy architecture, Fuzzy-SA algorithm, WDO algorithm, and Fuzzy-WDO algorithm are successfully employed for mobile robot navigation and obstacle avoidance in the different simulation and experimental environments. Firstly, the two simulation tests are conducted using all the developed techniques to compare their performances in the test environments. Then, the experiments have been done in the same environment to verify the effectiveness of the developed simulations. In the comparison study between the simulation and experiment results errors are recorded, and the errors are found due to the effect of slippage and friction between the wheels of the robot and surface during navigation in real time mode.

During experiment utmost care has been taken to minimize the slippage and friction between the wheels and surface. Still the effect of slippage and friction are unavoidable, and errors are recorded during the comparison of the results for travelling path length (6%) and for navigation time (7%).

After summarizing the above simulation and experimental studies, it can be concluded that the Fuzzy-Wind Driven Optimization algorithm gives the better results (in terms of path length and navigation time) as compared to rest of the techniques, which verifies the superiority of this technique.

Chapter 10

Conclusion and Scope for Future Research

10.1 Introduction

In this proposed research work, various soft computing techniques such as Hybrid Fuzzy (H-Fuzzy) architecture, Cascade Neuro-Fuzzy (CN-Fuzzy) architecture, Fuzzy-Simulated Annealing (Fuzzy-SA) algorithm, Wind Driven Optimization (WDO) algorithm, and Fuzzy-Wind Driven Optimization (Fuzzy-WDO) algorithm have been successfully designed and implemented for mobile robot navigation and obstacle avoidance in the different static and dynamic environments. This chapter summarizes the significant contributions, conclusion, and scope for the future research from proposed research work. Rest of the chapter is organized as follows: Section 10.2 presents the important contributions of the proposed dissertation. Section 10.3 demonstrates the conclusions of all chapters. Section 10.4 depicts the scope for future research from the present research.

10.2 Important contributions

The significant contributions of the dissertation are summarized as follows: -

- The kinematic and dynamic equations of the mobile robot have been described in the third chapter, which helps to control the motion and orientation of the robot in the environment.
- The Hybrid Fuzzy (H-Fuzzy) architecture, Cascade Neuro-Fuzzy (CN-Fuzzy) architecture, Fuzzy-Simulated Annealing (Fuzzy-SA) algorithm have been applied in this research work to improve the navigation and obstacle avoidance performance of the mobile robot in various (static and dynamic) environments.

- According to literature survey, this is for the first time that Wind Driven Optimization (WDO) algorithm has been applied for the mobile robot navigation and obstacle avoidance in the static and dynamic environments.
- Besides, this newly developed WDO algorithm is integrated with the fuzzy controller to adjust and optimize the antecedent and consequent parameters of the fuzzy membership function.
- The performance of these developed techniques are demonstrated through computer simulations using MATLAB software and implemented in real time by using Arduino microcontroller based experimental mobile robots.

10.3 Conclusions

According to the results obtained from the simulation and experimental studies using all the developed soft computing techniques, the major conclusions of the dissertation are summarized below: -

- The kinematic and dynamic analysis of the mobile robot have been presented in the third chapter, which is important for controlling the right and left wheel velocities of the mobile robot during navigation and obstacle avoidance in the environment.
- Chapter four discusses the design and implementation of the hybrid fuzzy (H-Fuzzy) architecture for mobile robot navigation in the presence of static and dynamic obstacles. The proposed H-Fuzzy architecture aims to control the turning angle (between the robot and goal) and motor velocities of a mobile robot from starting point to goal point with obstacle avoidance competence. This architecture is tested in various simulation and experimental environments and is found to be a good agreement.
- In chapter five, the Cascade Neuro-Fuzzy (CN-Fuzzy) architecture has been presented for intelligent navigation control of a mobile robot in the static and dynamic environments filled with obstacles. The cascade neural network is used to train the robot to reach the goal in the environment, and the fuzzy logic architecture is integrated with this cascade neural network to control the right and left motor velocities of the robot to protect it from the obstacles.

- In chapter six, a Takagi-Sugeno fuzzy model is integrated with the Simulated Annealing algorithm called as Fuzzy-Simulated Annealing (Fuzzy-SA) algorithm to optimize the navigation path length of the mobile robot in the given environment. The proposed algorithm receives the obstacle distance (inputs) from the group of sensors for selecting the suitable steering angle (output) control command for the mobile robot during navigation. Successful simulation and experimental studies verify the effectiveness and efficiency of this proposed Fuzzy-SA algorithm.
- Chapter seven and eight applies the Wind Driven Optimization (WDO) algorithm, and Fuzzy-Wind Driven Optimization (Fuzzy-WDO) to solve the optimal path planning problems of a mobile robot in different simulation and experimental environments. The WDO (Wind Driven Optimization) algorithm has been used to optimize and tune the input/output membership function parameters of the fuzzy controller. These algorithms improve the navigation performance of the mobile robot in the given environments and produce a smooth navigation path within a reasonable time.
- In chapter nine, the two simulation and experimental tests (test1 and test2) are conducted to compare the performance of all the developed soft computing techniques in the given environments. From the comparison tests, it is found that as compared to the H-Fuzzy, CN-Fuzzy, Fuzzy-SA, and WDO the Fuzzy-WDO provides better results (in terms of path length and navigation time) because WDO algorithm adjusts the membership function of the fuzzy controller, which verifies the superiority of this newly developed technique.
- During comparison studies between simulation and experiment, the average percentage of errors are found to be within 6% in terms of travelling path length and 7% in terms of navigation time.
- All the simulation results, experimental studies, and comparison with previous works (e.g., [11] and [45]) illustrate that the proposed soft computing techniques are indeed effective and feasible for navigation of mobile robot form start to goal in a cluttered environment.

10.4 Scope for Future Research

The present research work can be extended in a number of ways and some of these are listed below: -

- The proposed techniques are developed and tested for the navigation of single robot in dynamic environments. In future, these developed techniques will be extended for multi-robot navigation and obstacle avoidance in the dynamic environments.
- These developed soft computing techniques can be hybridized with other nature-inspired optimization algorithms such as Teaching-Learning-Based Optimization (TLBO), Firefly Algorithm (FA), Cuckoo Search (CS) algorithm, and Bat Algorithm, etc. to compare the navigational performances and obstacle avoidance strategy of the mobile robot.
- The proposed techniques may be implemented for the multiple swarms mobile robots navigation and obstacle avoidance with moving goal problems in the environment.
- The vision sensors like camera may be attached to the experimental mobile robot to improve the sensory system of the mobile robot for the betterment of the navigation accuracy.
- In future research work, these developed techniques will be tested in the rough terrain to study its performance.

Bibliography

[1] Ni J., Wu L., Fan X. and Yang S. X. Bioinspired Intelligent Algorithm and Its Applications for Mobile Robot Control: A Survey. *Computational Intelligence and Neuroscience*, Article ID 3810903: 1--15, 2016.

[2] Takahashi O. and Schilling R.J. Motion Planning in a Plane Using Generalized Voronoi Diagrams. *IEEE Transactions on Robotics and Automation*, 5(2): 143--150, 1989.

[3] Bhattacharya P. and Gavrilova M. L. Roadmap-Based Path Planning-Using the Voronoi Diagram for a Clearance-Based Shortest Path. *IEEE Robotics and Automation Magazine*, 15(2): 58--66, 2008.

[4] Gomez E. J., Martinez Santa F. and Martinez Sarmiento F. H. A Comparative Study of Geometric Path Planning Methods for a Mobile Robot: Potential Field and Voronoi Diagrams. In *IEEE International Congress of Engineering Mechatronics and Automation (CIIMA)*, Bogota, Colombia, pages 1--6, 2013.

[5] Abiyev R., Ibrahim D. and Erin B. Navigation of Mobile Robots in the Presence of Obstacles. *ELSEVIER Advances in Engineering Software*, 41(10): 1179--1186, 2010.

[6] Soltani A. R., Tawfik H., Goulermas J. Y. and Fernando T. Path Planning in Construction Sites: Performance Evaluation of the Dijkstra, A∗, and GA Search Algorithms. *ELSEVIER Advanced Engineering Informatics*, 16(4): 291--303, 2002.

[7] Masehian E. and Amin-Naseri M. R. A Voronoi Diagram–Visibility Graph–Potential Field Compound Algorithm for Robot Path Planning. *Journal of Robotic System*, 21(6): 275--300, 2004.

[8] Weigl M., Siemiaatkkowska B., Sikorski K. A. and Borkowski A. Grid-Based Mapping for Autonomous Mobile Robot. *ELSEVIER Robotics and Autonomous Systems*, 11(1): 13--21, 1993.

[9] Park K. H., Kim Y. J. and Kim J. H. Modular Q-Learning based Multi-Agent Cooperation for Robot Soccer. *ELSEVIER Robotics and Autonomous Systems*, 35(2): 109--122, 2001.

[10] Montaner M. B. and Ramirez-Serrano A. Fuzzy Knowledge-Based Controller Design for Autonomous Robot Navigation. *ELSEVIER Expert Systems with Applications*, 14(1): 179--186, 1998.

[11] Engedy I. and Horvath G. Artificial Neural Network based Local Motion Planning of a Wheeled Mobile Robot. In *IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 213--218, 2010.

[12] Zhu A. and Yang S. X. Neurofuzzy-Based Approach to Mobile Robot Navigation in Unknown Environments. *IEEE Transactions on Systems, Man, and Cybernetics Applications and Reviews*, 37(4): 610--621, 2007.

[13] Ghorbani A., Shiry S. and Nodehi A. Using Genetic Algorithm for a Mobile Robot Path Planning. In *IEEE International Conference on Future Computer and Communication*, Kuala Lumpur, Malaysia, pages 164--166, 2009.

[14] Ahmadzadeh S. and Ghanavati M. Navigation of Mobile Robot Using the PSO Particle Swarm Optimization. *Journal of Academic and Applied Studies (JAAS)*, 2(1): 32--38, 2012.

[15] Garcia M. P., Montiel O., Castillo O., Sepulveda R. and Melin P. Path Planning for Autonomous Mobile Robot Navigation with Ant Colony Optimization and Fuzzy Cost Function Evaluation. *ELSEVIER Applied Soft Computing*, 9(3): 1102--1110, 2009.

[16] Miao H. and Tian Y. C. Dynamic Robot Path Planning using an Enhanced Simulated Annealing Approach. *ELSEVIER Applied Mathematics and Computation*, 222: 420--437, 2013.

[17] Hui N. B., Mahendar V. and Pratihar D. K. Time-Optimal, Collision-Free Navigation of a Car-Like Mobile Robot using Neuro-Fuzzy Approaches. *ELSEVIER Fuzzy Sets and Systems*, 157(16): 2171--2204, 2006.

[18] Abadi D. N. M. and Khooban M. H. Design of Optimal Mamdani-Type Fuzzy Controller for Nonholonomic Wheeled Mobile Robots. *ELSEVIER Journal of King Saud University-Engineering Sciences*, 27(1): 92--100, 2015.

[19] Chakraborty N. and Ghosal A. Kinematics of Wheeled Mobile Robots on Uneven Terrain. *ELSEVIER Mechanism and Machine Theory*, 39(12): 1273--1287, 2004.

[20] Wang X. and Yang S. X. A Neuro-Fuzzy Approach to Obstacle Avoidance of a Nonholonomic Mobile Robot. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 29--34, 2003.

[21] Das T., Kar I. N. and Chaudhury S. Simple Neuron-Based Adaptive Controller for a Nonholonomic Mobile Robot Including Actuator Dynamics. *ELSEVIER Neurocomputing*, 69(16): 2140--2151, 2006.

[22] Martinez R., Castillo O. and Aguilar L. T. Optimization of Interval Type-2 Fuzzy Logic Controllers for a Perturbed Autonomous Wheeled Mobile Robot using Genetic Algorithms. *ELSEVIER Information Sciences*, 179(13): 2158--2174, 2009.

[23] Al-Araji A. S., Abbod M. F. and Al-Raweshidy H. S. Applying Posture Identifier in Designing an Adaptive Nonlinear Predictive Controller for Nonholonomic Mobile Robot. *ELSEVIER Neurocomputing*, 99: 543--554, 2013.

[24] Liang Y., Xu L., Wei R. and Hu H. Adaptive Fuzzy Control for Trajectory Tracking of Mobile Robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* pages 4755--4760, 2010.

[25] Zadeh L. A. The Concept of a Linguistic Variable and its Application to Approximate Reasoning-I. *ELSEVIER Information Sciences*, 8(3): 199--249, 1975.

[26] Ren L., Wang W. and Du Z. A New Fuzzy Intelligent Obstacle Avoidance Control Strategy for Wheeled Mobile Robot. In *IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1732--1737, 2012.

[27] Yousfi N., Rekik C., Jallouli M. and Derbel N. Optimized Fuzzy Controller for Mobile Robot Navigation in a Cluttered Environment. In *IEEE 7th International Multi-Conference on Systems, Signals and Devices*, pages 1--7, 2010.

[28] Qing-yong B., Shun-ming L., Wei-yan S. and Mu-jin A. A Fuzzy Behavior-Based Architecture for Mobile Robot Navigation in Unknown Environments. In *IEEE International Conference on Artificial Intelligence and Computational Intelligence*, pages 257--261, 2009.

[29] Boubertakh H., Tadjine M., Glorennec P. and Labiod S. A Simple Goal Seeking Navigation Method for a Mobile Robot using Human Sense, Fuzzy Logic and Reinforcement Learning. *Journal of Automatic Control, University of Belgrade*, 18(1): 23--27, 2008.

[30] Muthu T., Thierry Gloude R., Swaminathan S. and Satish Kumar L. Fuzzy Logic Controller for Autonomous Navigation. In *IEEE International Conference on Communications and Signal Processing (ICCSP)*, pages 81--92, 2012.

[31] El-Teleity S. A. L., Nossair Z. B., Mansour H. M. and TagElDein A. Fuzzy Logic Control of an Autonomous Mobile Robot. In *IEEE 16th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 188--193, 2011.

[32] Raguraman S. M., Tamilselvi D. and Shivakumar N. Mobile Robot Navigation Using Fuzzy logic Controller. In *IEEE 16th International Conference on Control, Automation, Communication and Energy Conservation*, pages 1--5, 2009.

[33] Wu S., Li Q., Zhu E., Xie J. and Zhichao G. Fuzzy Controller of Pipeline Robot Navigation Optimized by Genetic Algorithm. In *IEEE International Conference on Control and Decision Conference (CCDC)*, China, pages 904--908, 2008.

[34] Samsudin K., Ahmad F. A. and Mashohor S. A Highly Interpretable Fuzzy Rule Base using Ordinal Structure for Obstacle Avoidance of Mobile Robot. *ELSEVIER Applied Soft Computing*, 11(2): 1631--1637, 2011.

[35] Beom H. R. and Cho K. S. A Sensor-Based Navigation for a Mobile Robot using Fuzzy Logic and Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(3): 464--477, 1995.

[36] Pradhan S. K., Parhi D. R. and Panda A. K. Fuzzy Logic Techniques for Navigation of Several Mobile Robots. *ELSEVIER Applied Soft Computing*, 9(1): 290--304, 2009.

[37] Liu Q., Lu Y. G. and Xie C. X. Optimal Genetic Fuzzy Obstacle Avoidance Controller of Autonomous Mobile Robot Based on Ultrasonic Sensors. In *IEEE International Conference on Robotics and Biomimetics*, China, pages 125--129, 2006.

[38] Farooq U., Hasan K. M., Abbas G. and Asad M. U. Comparative Analysis of Zero Order Sugeno and Mamdani Fuzzy Logic Controllers for Obstacle Avoidance Behavior in Mobile Robot Navigation. In *IEEE International Workshop on Current Trends in Information Technology (CTIT)*, pages 113--119, 2011.

[39] Algabri M., Mathkour H., Ramdane H. and Alsulaiman M. Comparative Study of Soft Computing Techniques for Mobile Robot Navigation in an Unknown Environment. *ELSEVIER Computers in Human Behavior*, 50: 42--56, 2015.

[40] Hui N. B. and Pratihar D. K. A Comparative Study on Some Navigation Schemes of a Real Robot Tackling Moving Obstacles. *ELSEVIER Robotics and Computer-Integrated Manufacturing*, 25(4): 810--828, 2009.

[41] Rusu C. G., Birou I. T. and Szoke E. Fuzzy Based Obstacle Avoidance System for Autonomous Mobile Robot. In *IEEE International Conference on Automation Quality and Testing Robotics (AQTR)*, pages 1--6, 2010.

[42] Abdessemed F., Benmahammed K. and Monacelli E. A Fuzzy-Based Reactive Controller for a Non-Holonomic Mobile Robot. *ELSEVIER Robotics and Autonomous Systems*, 47(1): 31--46, 2004.

[43] Selekwa M. F., Dunlap D. D., Shi D. and Collins E. G. Robot Navigation in Very Cluttered Environments by Preference-Based Fuzzy Behaviors. *ELSEVIER Robotics and Autonomous Systems*, 56(3): 231--246, 2008.

[44] Pratihar D. K., Deb K. and Ghosh A. A Genetic-Fuzzy Approach for Mobile Robot Navigation among Moving Obstacles. *ELSEVIER International Journal of Approximate Reasoning*, 20(2): 145--172, 1999.

[45] Faisal M., Hedjar R., Al Sulaiman M. and Al-Mutib K. Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment. *INTECH International Journal of Advanced Robotic Systems*, 10(37): 1--7, 2013.

[46] Babalou A. and Seifiour N. Application of Fuzzy Decision Making in Mobile Robot Navigation in Dynamic Environments. In *IEEE International Conference on Fuzzy Systems*, Korea, pages 877--881, 2009.

[47] Li T. H. S., Chang S. J. and Chen Y. X. Implementation of Human-Like Driving Skills by Autonomous Fuzzy Behavior Control on an FPGA-Based Car-Like Mobile Robot. *IEEE Transactions on Industrial Electronics*, 50(5): 867--880, 2003.

[48] Li T. H. S., Chang S. J. and Tong W. Fuzzy Target Tracking Control of Autonomous Mobile Robots by Using Infrared Sensors. *IEEE Transactions on Fuzzy Systems*, 12(4): 491--501, 2004.

[49] Dongshu W., Yusheng Z. and Wenjie S. Behavior-Based Hierarchical Fuzzy Control for Mobile Robot Navigation in Dynamic Environment. In *IEEE International Conference on Control and Decision Conference (CCDC)*, pages 2419--2424, 2011.

[50] Antonelli G., Chiaverini S. and Fusco G. A Fuzzy-Logic-Based Approach for Mobile Robot Path Tracking. *IEEE Transactions on Fuzzy Systems*, 15(2): 211--221, 2007.

[51] Ayari E., Hadouaj S. and Ghedira K. A Fuzzy Logic Method for Autonomous Robot Navigation in Dynamic and Uncertain Environment Composed with Complex Traps. In *IEEE 5th International Multi-conference on Computing in the Global Information Technology*, pages 18--23, 2010.

[52] Zou A. M., Hou Z. G., Fu S. Y. and Tan M. Neural Networks for Mobile Robot Navigation: a Survey. In: *Advances in Neural Networks-ISNN*, pages 1218--1226, 2006.

[53] Xiao H., Liao L. and Zhou F. Mobile Robot Path Planning Based on Q-ANN. In *IEEE International Conference on Automation and Logistics*, China, pages 2650--2654, 2007.

[54] Rai N., and Rai B. Neural Network based Closed loop Speed Control of DC Motor using Arduino Uno. *International Journal of Engineering Trends and Technology*, 4(2): 137--140, 2013.

[55] Patino H. D. and Carelli R. Neural Network-Based Optimal Control for Autonomous Mobile Vehicle Navigation. In *IEEE International Symposium on Intelligent Control*, pages 391--396, 2004.

[56] Yang S. X. and Meng M. Neural Network Approaches to Dynamic Collision-Free Trajectory Generation. *IEEE Transactions on Systems, Man, and Cybernetics Applications and Reviews*, 31(3): 302--318, 2001.

[57] Nichols E., McDaid L. J. and Siddique N. Biologically Inspired SNN for Robot Control. *IEEE Transactions on Cybernetics*, 15(2): 115--128, 2013.

[58] Motlagh O., Nakhaeinia D., Tang S. H., Karasfi B. and Khaksar W. Automatic Navigation of Mobile Robots in Unknown Environments. *SPRINGER Neural Computing and Applications*, 24(7-8): 1569--1581, 2014.

[59] Gavrilov A. V. and Lee S. An Architecture of Hybrid Neural Network Based Navigation System for Mobile Robot. In *IEEE Seventh International Conference on Intelligent Systems Design and Applications*, pages 587--590, 2007.

[60] Singh M. K. and Parhi D. R. Path Optimisation of a Mobile Robot Using an Artificial Neural Network Controller. *International Journal of Systems Science*, 42(1): 107--120, 2011.

[61] Rossomando F. G. and Soria C. M. Design and Implementation of Adaptive Neural PID for Non Linear Dynamics in Mobile Robots. *IEEE Latin America Transactions*, 13(4): 913--918, 2015.

[62] Al-Jarrah R., Shahzad A. and Roth H. Path Planning and Motion Coordination for Multi-Robots System Using Probabilistic Neuro-Fuzzy. *ELSEVIER IFAC-Papers Online*, 48(10): 46--51, 2015.

[63] Janglova D. Neural Networks in Mobile Robot Motion. *International Journal of Advanced Robotic Systems*, 1(1): 15--22, 2004.

[64] Glasius R., Komoda A. and Gielen S. C. Neural Network Dynamics for Path Planning and Obstacle Avoidance. *Neural Networks*, 8(1): 125--133, 1995.

[65] Kim C. J. and Chwa D. Obstacle Avoidance Method for Wheeled Mobile Robots Using Interval Type-2 Fuzzy Neural Network. *IEEE Transactions on Fuzzy Systems*, 23(3): 677--687, 2015.

[66] Mahmud F., Arafat A. and Zuhori S. T. Intelligent Autonomous Vehicle Navigated by Using Artificial Neural Network. In *IEEE International Conference on Electrical and Computer Engineering*, Dhaka, Bangladesh, pages 105--108, 2012.

[67] Chohra A., Farah A. and Benmehrez C. Neural Navigation Approach for Intelligent Autonomous Vehicles (IAV) in Partially Structured Environments. *SPRINGER Applied Intelligence*, 8(3): 219--233, 1998.

[68] Brahmi H., Ammar B. and Alimi A. M. Intelligent Path Planning Algorithm For Autonomous Robot Based on Recurrent Neural Networks. In *IEEE International Conference on Advanced Logistics and Transport (ICALT)*, pages 199--204, 2013.

[69] Yang S. X., Hu T., Yuan X., Liu P. X. and Meng M. A Neural Network based Torque Controller for Collision-Free Navigation of Mobile Robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, pages 13--18, 2003.

[70] Al Mutib K. and Mattar E. Neuro-fuzzy Controlled Autonomous Mobile Robotics System. In *IEEE 13th International Conference on Modelling and Simulation*, UK, pages 1--7, 2011.

[71] Godjevac J. and Steele N. Neuro-Fuzzy Control of a Mobile Robot. *ELSEVIER Neurocomputing*, 28(1): 127--143, 1999.

[72] Li W., Ma C. and Wahl F. M. A Neuro-Fuzzy System Architecture for Behavior-Based Control of a Mobile Robot in Unknown Environments. *ELSEVIER Fuzzy Sets and Systems*, 87(2): 133--140, 1997.

[73] Joshi M. M. and Zaveri M. Neuro-Fuzzy Based Autonomous Mobile Robot Navigation System. In *IEEE 11th International Conference Control, Automation, Robotics and Vision*, Singapore, pages 384--389, 2010.

[74] Marichal G. N., Acosta L., Moreno L., Mendez J. A., Rodrigo J. J. and Sigut M. Obstacle Avoidance for a Mobile Robot: A Neuro-Fuzzy Approach. *ELSEVIER Fuzzy Sets and Systems*, 124(2): 171--179, 2001.

[75] Zhang N., Beetner D., Wunsch D. C., Hemmelman B. and Hasan A. An Embedded Real-Time Neuro-Fuzzy Controller for Mobile Robot Navigation. In *IEEE International Conference on Fuzzy Systems*, pages 319--324, 2005.

[76] Baturone I., Gersnoviez A. and Barriga A. Neuro-Fuzzy Techniques to Optimize an FPGA Embedded Controller for Robot Navigation. *ELSEVIER Applied Soft Computing*, 21: 95--106, 2014.

[77] Ma X., Li X. and Qiao H. Fuzzy Neural Network-Based Real-Time Self-Reaction of Mobile Robot in Unknown Environments. *ELSEVIER Mechatronics*, 11(8): 1039--1052, 2001.

[78] Imen M., Mohammad M. and Shoorehdeli M. A. Tracking Control of Mobile Robot Using ANFIS. In *IEEE International Conference on Mechatronics and Automation*, Beijing, China, pages 422--427, 2011.

[79] Ganapathy V., Yun S. C. and Ng J. Fuzzy and Neural Controllers for Acute Obstacle Avoidance in Mobile Robot Navigation. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Singapore, pages 1236--1241, 2009.

[80] Zhao T. and Wang Y. A Neural-Network based Autonomous Navigation System using Mobile Robots. In *IEEE 12th International Conference on Control, Automation, Robotics & Vision (ICARCV)*, Guangzhou, China, pages 1101--1106, 2012.

[81] Kumar D. and Dhama K. Neuro-Fuzzy Control of an Intelligent Mobile Robot. In: *IEEE Second International Conference on Advanced Computing & Communication Technologies (ACCT)*, pages 106--111, 2012.

[82] Song K. T. and Sheen L. H. Heuristic Fuzzy-Neuro Network and its Application to Reactive Navigation of a Mobile Robot. *ELSEVIER Fuzzy Sets and Systems*, 110(3): 331--340, 2000.

[83] Lee C. H. and Chiu M. H. Recurrent Neuro Fuzzy Control Design for Tracking of Mobile Robots via Hybrid Algorithm. *ELSEVIER Expert Systems with Applications*, 36(5): 8993--8999, 2009.

[84] Deshpande S. U. and Bhosale S. S. Adaptive Neuro-Fuzzy Inference System based Robotic Navigation. In *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1--4, 2013.

[85] Rusu P., Petriu E. M., Whalen T. E., Cornell A. and Spoelder H. J. Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation. *IEEE Transactions on Instrumentation and Measurement*, 52(4): 1335--1340, 2003.

[86] Pothal J. K. and Parhi D. R. Navigation of Multiple Mobile Robots in a Highly Clutter Terrains using Adaptive Neuro-Fuzzy Inference System. *ELSEVIER Robotics and Autonomous Systems*, 72: 48--58, 2015.

[87] Ng K. C. and Trivedi M. M. A Neuro-Fuzzy Controller for Mobile Robot Navigation and Multirobot Convoying. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(6): 829--840, 1998.

[88] Demirli K. and Khoshnejad M. Autonomous Parallel Parking of a Car-Like Mobile Robot by a Neuro-Fuzzy Sensor-Based Controller. *ELSEVIER Fuzzy Sets and Systems*, 160(19): 2876--2891, 2009.

[89] Al-Mayyahi A., Wang W. and Birch P. Adaptive Neuro-Fuzzy Technique for Autonomous Ground Vehicle Navigation. *Robotics*, 3(4): 349--370, 2014.

[90] Pradhan S. K., Parhi, D. R. and Panda A. K. Neuro-Fuzzy Technique for Navigation of Multiple Mobile Robots. *SPRINGER Fuzzy Optimization and Decision Making*, 5(3): 255--288, 2006.

[91] Algabri M., Mathkour H. and Ramdane H. Mobile Robot Navigation and Obstacle-Avoidance Using ANFIS in Unknown Environment. *International Journal of Computer Applications*, 91(14): 36--41, 2014.

[92] Elshamli A., Abdullah H. A. and Areibi S. Genetic Algorithm for Dynamic Path Planning. In *IEEE International Conference on Electrical and Computer Engineering*, pages 677--680, 2004.

[93] Mohanta J. C., Parhi D. R. and Patel S. K. Path Planning Strategy for Autonomous Mobile Robot Navigation Using Petri-GA Optimisation. *ELSEVIER Computers and Electrical Engineering*, 37(6): 1058--1070, 2011.

[94] Kubota N., Morioka T., Kojima F. and Fukuda T. Learning of Mobile Robots Using Perception-Based Genetic Algorithm. *ELSEVIER Measurement*, 29(3): 237--248, 2001.

[95] Tuncer A. and Yildirim M. Dynamic Path Planning of Mobile Robots with Improved Genetic Algorithm. *ELSEVIER Computers and Electrical Engineering*, 38(6): 1564--1572, 2012.

[96] Ming L., Zailin G. and Shuzi Y. Mobile Robot Fuzzy Control Optimization Using Genetic Algorithm. *ELSEVIER Artificial Intelligence in Engineering*, 10(4): 293--298, 1996.

[97] Hu Y., Yang S. X., Xu L. Z. and Meng M. Q. A Knowledge based Genetic Algorithm for Path Planning in Unstructured Mobile Robot Environments. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 767--772, 2004.

[98] Liu Q., Lu Y. G. and Xie C. X. Fuzzy Obstacle-avoiding Controller of Autonomous Mobile Robot Optimized by Genetic Algorithm under Multi-obstacles Environment. In *IEEE 6th World Congress on Intelligent Control and Automation*, China, pages 3255--3259, 2006.

[99] Li Q., Zhang W., Yin Y., Wang Z. and Liu G. An Improved Genetic Algorithm of Optimum Path Planning for Mobile Robots. In *IEEE 6th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 637--642, 2006.

[100] Qu H., Xing K. and Alexander T. An Improved Genetic Algorithm with Co-Evolutionary Strategy for Global Path Planning of Multiple Mobile Robots. *ELSEVIER Neurocomputing,* 120: 509--517, 2013.

[101] Algabri M., Mathkour H., Hedjar R., Alsulaiman M. and Al Mutib K. Self-learning Mobile Robot Navigation in Unknown Environment Using Evolutionary Learning. *Journal of Universal Computer Science*, 20(10): 1459--1468, 2014.

[102] Castillo O., Trujillo L. and Melin P. Multiple Objective Genetic Algorithms for Path-Planning Optimization in Autonomous Mobile Robots. *SPRINGER Soft Computing*, 11(3): 269--279, 2007.

[103] Arora T., Gigras Y. and Arora V. Robotic Path Planning using Genetic Algorithm in Dynamic Environment. *International Journal of Computer Application*, 89(11): 8--12 2014.

[104] Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H. and Teller E. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6): 1087--1092, 1953.

[105] Yanar T. A. and Akyurek Z. Fuzzy Model Tuning Using Simulated Annealing. *ELSEVIER Expert Systems with Applications*, 38(7): 8159--8169, 2011.

[106] Chang C. C. and Song K. T. Sensor-Based Motion Planning of a Mobile Robot in a Dynamic Environment. In *IEEE International Conference on Industrial Electronics, Control, and Instrumentation*, pages 766--771, 1996.

[107] Martinez-Alfaro H. and Gomez-Garcia S. Mobile Robot Path Planning and Tracking Using Simulated Annealing and Fuzzy Logic Control. *ELSEVIER Expert Systems with Applications*, 15(3): 421--429, 1998.

[108] Zhu Q., Yan Y. and Xing Z. Robot Path Planning Based on Artificial Potential Field Approach with Simulated Annealing. In *IEEE Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 622--627, 2006.

[109] Precup R. E., Radac M. B., Dragos C. A., Preitl S. and Petriu E. M. Simulated Annealing Approach to Fuzzy Modeling of Servo Systems. In *IEEE International Conference on Cybernetics*, pages 267--272, 2013.

[110] Janabi-Sharifi F. and Vinke D. Integration of the Artificial Potential Field Approach with Simulated Annealing for Robot Path Planning. In *IEEE International Symposium on Intelligent Control*, pages 536--541, 1993.

[111] Tavares R. S., Martins T. C. and Tsuzuki M. D. S. G. Simulated Annealing with Adaptive Neighborhood: A Case Study in Off-Line Robot Path Planning. *ELSEVIER Expert Systems with Applications*, 38(4): 2951--2965, 2011.

[112] Liang Y. and Xu L. Global Path Planning for Mobile Robot based Genetic Algorithm and Modified Simulated Annealing Algorithm. In *ACM Proceedings of Summit on Genetic and Evolutionary Computation*, pages 303--308, 2009.

[113] Nakamura E. and Kehtarnavaz N. Optimization of Fuzzy Membership Function Parameters. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1--6, 1995.

[114] Hussein A., Mostafa H., Badrel-din M., Sultan O. and Khamis A. Metaheuristic Optimization Approach to Mobile Robot Path Planning. In *IEEE International Conference on Engineering and Technology (ICET)*, pages 1--6, 2012.

[115] Miao H. and Tian Y. C. Robot Path Planning in Dynamic Environments using a Simulated Annealing Based Approach. In *IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1253--1258, 2008.

[116] Zhang Q., Ma J. and Liu Q. Path Planning based Quadtree Representation for Mobile Robot Using Hybrid-Simulated Annealing and Ant Colony Optimization Algorithm. In *IEEE International World Congress on Intelligent Control and Automation*, China, pages 2537--2542, 2012.

[117] Gao M. and Tian J. Path Planning for Mobile Robot Based on Improved Simulated Annealing Artificial Neural Network. In *IEEE Third International Conference on Natural Computation*, pages 8--12, 2007.

[118] Synodinos A. and Aspragathos N. A. Robot Path Planning of a Mobile Robot using Solid Modeling Techniques on Potential Fields. In *IEEE International Conference on Mechatronics and Embedded Systems and Applications (MESA)*, pages 549--553, 2010.

[119] Zhao Y. and Zu W. Real-Time Obstacle Avoidance Method for Mobile Robots Based on a Modified Particle Swarm Optimization. In *IEEE International Joint Conference on Computational Sciences and Optimization (CSO)*, pages 269--272, 2009.

[120] Castillo O., Martinez-Marroquin R., Melin P., Valdez F. and Soria J. Comparative Study of Bio-Inspired Algorithms Applied to the Optimization of Type-1 and Type-2 Fuzzy Controllers for an Autonomous Mobile Robot. *ELSEVIER Information sciences*, 192: 19--38, 2012.

[121] Zhang Y., Gong D. W. and Zhang J. H. Robot Path Planning in Uncertain Environment using Multi-Objective Particle Swarm Optimization. *ELSEVIER Neurocomputing*, 103: 172--185, 2013.

[122] Zhang Q. and Li S. A Global Path Planning Approach based on Particle Swarm Optimization for a Mobile Robot. In *International Conference on Robotics, Control & Manufacturing Technology World Scientific and Engineering Academy and Society (WSEAS)*, pages 263--267, 2007.

[123] Raja P. and Pugazhenthi S. Path Planning for Mobile Robots in Dynamic Environments using Particle Swarm Optimization. In *IEEE International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom)*, pages 401--405, 2009.

[124] Masehian E. and Sedighizadeh D. A Multi-Objective PSO-based Algorithm for Robot Path Planning. In *IEEE International Conference on Industrial Technology (ICIT)*, pages 465--470, 2010.

[125] Wong C., Wang H. and Li S. PSO-based Motion Fuzzy Controller Design for Mobile Robots. *International Journal of Fuzzy Systems*, 10(1): 284--292, 2008.

[126] Li Q., Tang Y., Wang L., Zhang C. and Yin Y. A Specialized Particle Swarm Optimization for Global Path Planning of Mobile Robots. In *IEEE Third International Workshop on Advanced Computational Intelligence (IWACI)*, pages 271--276, 2010.

[127] Huang H. C. FPGA-Based Parallel Metaheuristic PSO Algorithm and its Application to Global Path Planning for Autonomous Robot Navigation. *SPRINGER Journal of Intelligent & Robotic Systems*, 76(3-4): 475--488, 2014.

[128] Chung H. Y., Hou C. C. and Liu S. C. Automatic Navigation of a Wheeled Mobile Robot using Particle Swarm Optimization and Fuzzy Control. In *IEEE International Symposium on Industrial Electronics (ISIE)*, pages 1--6, 2013.

[129] Shiltagh N. A. and Jalal L. D. Optimal Path Planning for Intelligent Mobile Robot Navigation using Modified Particle Swarm Optimization. *International Journal of Engineering and Advanced Technology*, 2(4): 260--267, 2013.

[130] Chatterjee A. and Matsuno F. A Geese PSO Tuned Fuzzy Supervisor for EKF based Solutions of Simultaneous Localization and Mapping (SLAM) Problems in Mobile Robots. *ELSEVIER Expert Systems with Applications*, 37(8): 5542--5548, 2010.

[131] Juang C. F. and Chang Y. C. Evolutionary-Group-based Particle-Swarm Optimized Fuzzy Controller with Application to Mobile-Robot Navigation in Unknown Environments. *IEEE Transactions on Fuzzy Systems*, 19(2): 379--392, 2011.

[132] Lu L. and Gong D. Robot Path Planning in Unknown Environments using Particle Swarm Optimization. In *IEEE International Conference on Natural Computation (ICNC)*, pages 422--426, 2008.

[133] Allawi Z. T. and Abdalla T. Y. A PSO-Optimized Type-2 Fuzzy Logic Controller for Navigation of Multiple Mobile Robots. In *IEEE International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 33--39, 2014.

[134] Dorigo M. and Caro G. D. Ant Colony Optimization: A New Meta-Heuristic. In *IEEE International Congress on Evolutionary Computation*, pages 1470--1477, 1999.

[135] Guan-Zheng T. A. N., Huan H. and Sloman A. Ant Colony System Algorithm for Real-Time Globally Optimal Path Planning of Mobile Robots. *ELSEVIER Acta Automatica Sinica*, 33(3): 279--285, 2007.

[136] Purian F. K. and Sadeghian E. Mobile Robots Path Planning using Ant Colony Optimization and Fuzzy Logic Algorithms in Unknown Dynamic Environments. In *IEEE International Conference on Control, Automation, Robotics and Embedded Systems (CARE)*, pages 1--6, 2013.

[137] Bi Z., Yimin Y. and Yisan X. Mobile Robot Navigation in Unknown Dynamic Environment Based on Ant Colony Algorithm. In *IEEE WRI Global Congress on Intelligent Systems*, pages 98--102, 2009.

[138] Dong J., Liu B., Peng K. and Yin Y. Robot Obstacle Avoidance based on an Improved Ant Colony Algorithm. In *IEEE WRI Global Congress on Intelligent Systems*, pages 103--106, 2009.

[139] Ganapathy V., Jie T. T. J. and Parasuraman S. Improved Ant Colony Optimization for Robot Navigation. In *IEEE International Symposium on Mechatronics and its Applications (ISMA)*, pages 1--6, 2010.

[140] Fan X., Luo X., Yi S., Yang S. and Zhang H. Optimal Path Planning for Mobile Robots Based on Intensified Ant Colony Optimization Algorithm. In *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, pages 131--136, 2003.

[141] Sariff N. B. and Buniyamin N. Comparative Study of Genetic Algorithm and Ant Colony Optimization Algorithm Performances for Robot Path Planning in Global Static Environments of Different Complexities. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 132--137, 2009.

[142] Hsu C. C., Hou R. Y. and Wang W. Y. Path Planning for Mobile Robots Based on Improved Ant Colony Optimization. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2777--2782, 2013.

[143] Ganganath N., Cheng C. T. and Tse C. K. An ACO-based Off-line Path Planner for Nonholonomic Mobile Robots. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1038--1041, 2014.

[144] Juang C. F. and Hsu C. H. Reinforcement Ant Optimized Fuzzy Controller for Mobile-Robot Wall-Following Control. *IEEE Transactions on Industrial Electronics*, 56(10): 3931--3940, 2009.

[145] Hsu C. H. and Juang C. F. Evolutionary Robot Wall-Following Control using Type-2 Fuzzy Controller with Species-DE-Activated Continuous ACO. *IEEE Transactions on Fuzzy Systems*, 21(1): 100--112, 2013.

[146] Juang C. F., Lai M. G. and Zeng W. T. Evolutionary Fuzzy Control and Navigation for Two Wheeled Robots Cooperatively Carrying an Object in Unknown Environments. *IEEE Transactions on Cybernetics*, 45(9): 1731--1743, 2015.

[147] Hsu C. H. and Juang C. F. Multi-Objective Continuous-Ant-Colony-Optimized FC for Robot Wall-Following Control. *IEEE Computational Intelligence Magazine*, 8(3): 28--40, 2013.

[148] Chen X., Kong Y., Fang X., and Wu Q. A Fast Two–Stage ACO Algorithm for Robotic Path Planning. *SPRINGER Neural Computing and Applications*, 22(2): 313--319, 2013.

[149] Hossain M. A., and Ferdousand I. Autonomous Robot Path Planning in Dynamic Environment using a New Optimization Technique Inspired by Bacterial Foraging Technique. *ELSEVIER Robotics and Autonomous Systems*, 64: 137--141, 2015.

[150] Liang X. D., Li L. Y., Wu J. G. and Chen H. N. Mobile Robot Path Planning based on Adaptive Bacterial Foraging Algorithm. *SPRINGER Journal of Central South University,* 20: 3391--3400, 2013.

[151] Brand M. and Yu X. H. Autonomous Robot Path Optimization using Firefly Algorithm. In *IEEE International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 1028--1032. 2013.

[152] Mohajer B., Kiani K., Samiei E. and Sharifi M. A New Online Random Particles Optimization Algorithm for Mobile Robot Path Planning in Dynamic Environments. *Hindawi Mathematical Problems in Engineering*, Article ID 491346: 1--9, 2013.

[153] Bayraktar Z., Komurcu M., Bossard J. A. and Werner D. H. The Wind Driven Optimization Technique and its Application in Electromagnetics. *IEEE Transactions on Antennas and Propagation*, 61(5): 2745--2757, 2013.

[154] Bayraktar Z., Komurcu M., Jiang Z. H., Werner D. H. and Werner P. L. Stub-loaded Inverted-F Antenna Synthesis via Wind Driven Optimization. In *IEEE International Symposium on Antennas and Propagation (APSURSI)*, pages 2920--2923, 2011.

[155] Bayraktar Z., Turpin J. P. and Werner D. H. Nature-Inspired Optimization of High-Impedance Metasurfaces with Ultrasmall Interwoven Unit Cells. *IEEE Antennas and Wireless Propagation Letters*, 10: 1563--1566, 2011.

[156] Kuldeep B., Singh V. K., Kumar A. and Singh G. K. Design of Two-Channel Filter Bank Using Nature Inspired Optimization based Fractional Derivative Constraints. *ELSEVIER ISA Transactions*, 54: 101--116, 2015.

[157] Bhandari A. K., Singh V. K., Kumar A. and Singh G. K. Cuckoo Search Algorithm and Wind Driven Optimization based Study of Satellite Image Segmentation for Multilevel Thresholding using Kapur's Entropy. *ELSEVIER Expert Systems with Applications,* 41(7): 3538--3560, 2014.

[158] Hochsteiner De Vasconcelos Segundo E., Levati Amoroso A., Cocco Mariani V. and Dos Santos Coelho L. A Wind Driven Approach Using Levy Flights for Global Continuous

Optimization. In *IEEE International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, pages 75--80, 2014.

[159] Boulesnane A. and Meshoul S. A New Multi-Region Modified Wind Driven Optimization Algorithm with Collision Avoidance for Dynamic Environments. In *Proceedings of Advances in Swarm Intelligence*, Springer, pages 412—421, 2014.

[160] Mohareri O., Dhaouadi R. and Rad A. B. Indirect Adaptive Tracking Control of a Nonholonomic Mobile Robot via Neural Networks. *Elsevier Neurocomputing*, 88: 54--66, 2012.

[161] Hanafi D., Abueejela Y. M. and Zakaria M. F. Wall Follower Autonomous Robot Development Applying Fuzzy Incremental Controller. *Intelligent Control and Automation*, 4(1): 18--25, 2013.

[162] Fierro R. and Lewis F. L. Control of a Nonholonomic Mobile Robot Using Neural Networks. *IEEE Transactions on Neural Networks*, 9(4): 589--600, 1998.

[163] Cherroun L. and Boumehraz M. Fuzzy Behavior Based Navigation Approach for Mobile Robot in Unknown Environment. *Journal of Electrical Engineering*, 13(4): 1--8, 2013.

[164] Chayjan R. A. and Ashari M. E. Modeling Isosteric Heat of Soya Bean for Desorption Energy Estimation using Neural Network Approach. *Chilean Journal of Agricultural Research*, 70(4): 616--625, 2010.

[165] Begum M., Mann G. K. and Gosine R. G. Integrated Fuzzy Logic and Genetic Algorithmic Approach for Simultaneous Localization and Mapping of Mobile Robots. *ELSEVIER Applied Soft Computing*, 8(1): 150--165, 2008.

[166] Contreras-Cruz M. A., Ayala-Ramirez V. and Hernandez-Belmonte U. H. Mobile Robot Path Planning using Artificial Bee Colony and Evolutionary Programming. *ELSEVIER Applied Soft Computing* 30: 319--328, 2015.

[167] Mohanty P. K. and Parhi D. R. Optimal Path Planning for a Mobile Robot using Cuckoo Search Algorithm. *TAYLOR & FRANCIS Journal of Experimental & Theoretical Artificial Intelligence*, 28(1-2): 35--52, 2014.

[168] Vakil-Baghmisheh M. T. and Navarbaf A. A Modified Very Fast Simulated Annealing Algorithm. In *IEEE International Symposium on Telecommunications*, pages 61--66, 2008.

[169] Jianguo W., Yilong Z. and Linlin X. Adaptive Genetic Algorithm Enhancements for Path Planning of Mobile Robots. In *IEEE International Conference on Measuring Technology and Mechatronics Automation*, pages 416--419, 2010.

[170] Liu C., Gao Z. and Zhao W. A New Path Planning Method Based on Firefly Algorithm. In *IEEE Fifth International Joint Conference on Computational Sciences and Optimization (CSO)*, pages 775--778, 2012.

[171] Motlagh O. R. E., Hong T. S. and Ismail N. Development of a New Minimum Avoidance System for a Behavior-Based Mobile Robot. *ELSEVIER Fuzzy Sets and Systems*, 160(13): 1929--1946, 2009.

[172] Mohanty P. K. and Parhi D. R. Navigation of Autonomous Mobile Robot Using Adaptive Network Based Fuzzy Inference System. *SPRINGER Journal of Mechanical Science and Technology*, 28(7): 2861--2868, 2014.

[173] Deepak B. B. V. L., Parhi D. R. and Raju B. M. V. A. Advance Particle Swarm Optimization-Based Navigational Controller for Mobile Robot. *SPRINGER Arabian Journal for Science and Engineering*, 39(8): 6477--6487, 2014.

[174] Tahmasebi P. and Hezarkhani A. A Hybrid Neural Networks-Fuzzy Logic-Genetic Algorithm for Grade Estimation. *ELSEVIER Computers & Geosciences*, 42: 18--27, 2014.

Dissemination

Journal Articles

1. **Anish Pandey**, and Dayal R. Parhi, “Multiple Mobile Robots Navigation and Obstacle Avoidance Using Minimum Rule Based ANFIS Network Controller in the Cluttered Environment”, *SOJ Robotics Automation*, 1(1): 1--11, 2016.

2. **Anish Pandey** and Dayal R. Parhi “New Algorithm for Behaviour-Based Mobile Robot Navigation in Cluttered Environment Using Neural Network Architecture”, *World Journal of Engineering*, 13(2): 129--141, 2016.

3. **Anish Pandey** and Dayal R. Parhi “Autonomous Mobile Robot Navigation in Cluttered Environment using Hybrid Takagi-Sugeno Fuzzy Model and Simulated Annealing Algorithm Controller”, *World Journal of Engineering*, Accepted, Paper ID-wje20160115003.

4. Dayal Ramakrushna Parhi, and **Anish Pandey**, “A Review of Mobile Robot Localization and Navigation”, *International Science Press IJAAIES*, 4(2): 103--109, 2012.

5. Dayal R. Parhi, **Anish Pandey**, and Rakesh Kumar Sonkar, “A Review of Mobile Robot Navigation Using a Fuzzy Logic”, *International Science Press IJAICR*, 5(2): 103--108, 2013.

Conference Presentations

1. **Anish Pandey**, Rakesh Kumar Sonkar, Krishna Kant Pandey, and D. R. Parhi, “Path Planning Navigation of Mobile Robot with Obstacles Avoidance Using Fuzzy Logic Controller”, In *IEEE 8th International Conference on Intelligent System and Control (ISCO)*, Coimbatore, pages 36--41, 2014.

2. **Anish Pandey**, and Dayal Ramakrushna Parhi, “MATLAB Simulation for Mobile Robot Navigation with Hurdles in Cluttered Environment Using Minimum Rule Based Fuzzy Logic Controller”, In *ELSEVIER Procedia Technology*, 2014, Vol. 14, pages 28--34.

3. **Anish Pandey**, Rakesh Kumar Sonkar, P. K. Mohanty, and D. R. Parhi, “Navigation of Mobile Robot with Obstacles Avoidance Using Fuzzy Logic”, In *Proceedings of*

International Conference on Advances in Mechanical Engineering, COEP, Pune, Maharashtra, Paper ID ICAME-S1/P1, May 29-31, 2013.

4. Prases K. Mohanty, Dayal R. Parhi, Alok K. Jha, and **Anish Pandey**, "Path Planning of an Autonomous Mobile Robot using Adaptive Network Based Fuzzy Controller", In *IEEE 3rd International Conference on Advance Computing Conference (IACC)*, pages 651--656, 2013.

5. **Anish Pandey**, Saroj Kumar, Krishna Kant Pandey, Dayal R. Parhi, "Mobile robot navigation in unknown static environments using ANFIS controller", In: *ELSEVIER Perspectives in Science*, dx.doi.org/10.1016/j.pisc.2016.04.094, 2016.

Communicated Articles

1. **Anish Pandey** and Dayal R. Parhi, "Mobile Robot Optimum Path Planning in Cluttered Environment using Wind Driven Optimization Algorithm", *SPRINGER Soft Computing*, Article No. **SOCO-D-15-00890**.

2. Dayal R. Parhi and **Anish Pandey**, "Optimum Path Planning of Mobile Robot in Unknown Static and Dynamic Environments using Fuzzy-Wind Driven Optimization Algorithm", *SPRINGER Intelligent Service Robotics*, Article No. **JIST-D-16-00018**.

Patent

1. Automated Four-Wheeled Intelligent Stair Climbing Mobile Robot (**Application No.: 201631009662**) Inventors: 1. Dr. Dayal R. Parhi, 2. Mr. **Anish Pandey** (filed).

Vitae

Mr. **Anish Pandey** was born in 1988 at Bilaspur district of Chhattisgarh (C.G.). He has completed his class X and class XII from G. T. B. Higher Secondary School, Bilaspur (C.G.) in 2003 and 2005, respectively. He completed his Bachelor of Engineering in Industrial and Production Engineering from Guru Ghasidas Central University, Bilaspur, in 2009 and his Master of Engineering in Mechanical Engineering (specialization in machine design) from Chhattisgarh Swami Vivekananda Technical University, Bhilai, in 2011. After the completion of his Post-Graduation he has joined Ph.D. program at Department of Mechanical Engineering, National Institute of Technology, Rourkela in July 2012 and submitted the Ph.D. thesis in April 2016. His research interest includes Mobile Robot Navigation, Humanoid Robots, and Autonomous Aerial Robots.

Appendix-A

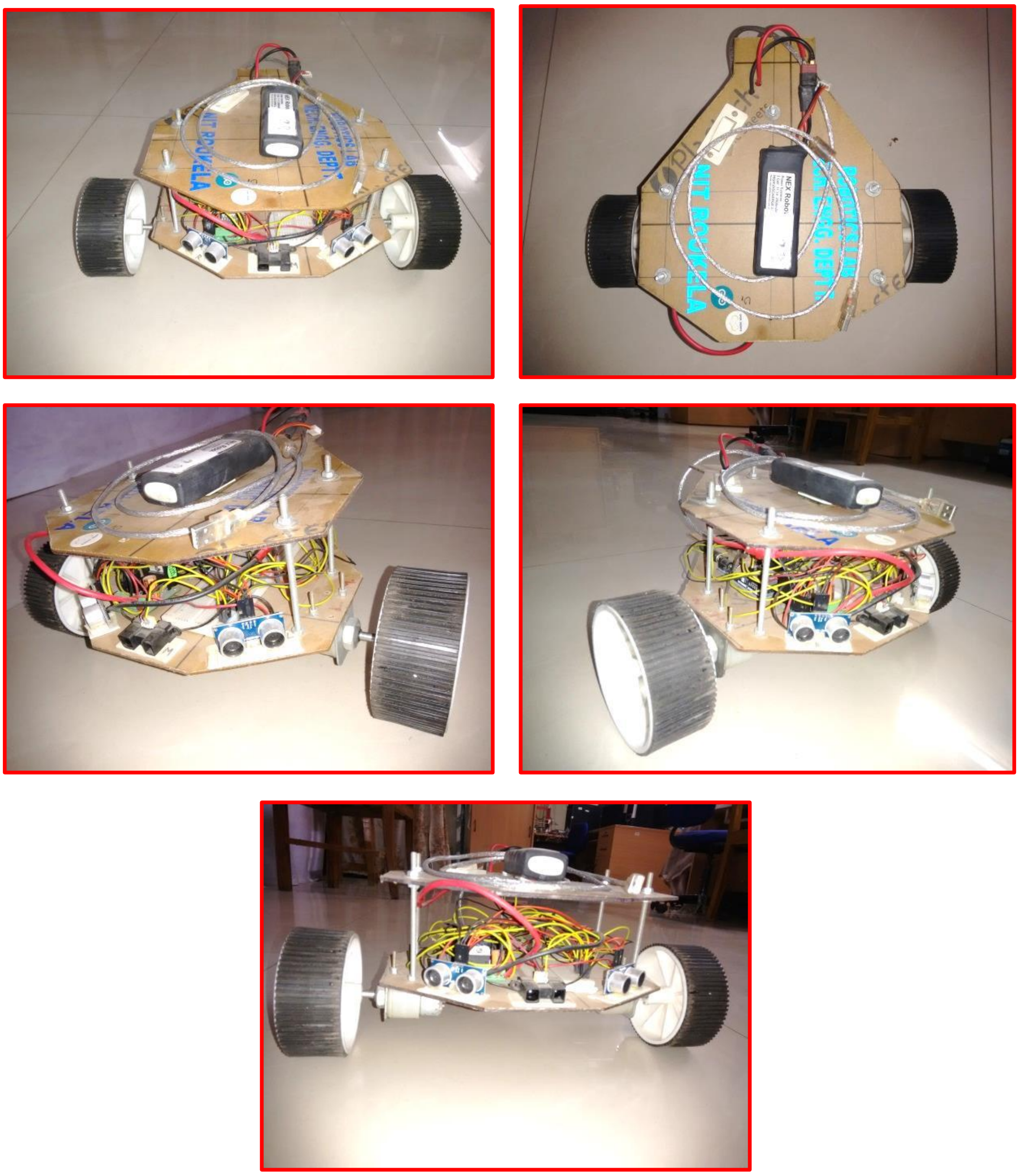


Figure A1: Arduino microcontroller based experimental differential drive mobile robot.

Table A1: Specifications of the experimental mobile robot

Name	Specifications
Microcontroller	Arduino MEGA 2560 (ATmega2560).
Flash Memory	256 KB (ATmega2560).
Operating Voltage	5V.
SRAM	8 KB (ATmega2560).
Input Voltage	7–12V (Recommended).
Input Voltage (Limits)	6–20V.
Digital Input/Output Pins	54 (of Which 15 Can be Used as PWM Outputs).
Analog Input Pins	16.
Motors	2 DC, 30RPM Centre Shaft Economy Series DC Motor.
Motors Driver	L298, Up to 46V, 2A Dual DC Two Motor Drivers.
Motor Speed	Max: 30RPM, Min: 12RPM.
Wheel	Wheel Diameter: 106mm, Wheel Thickness: 44mm, Hole Diameter: 8mm.
Sensors	One Sharp Infrared Range Sensor (GP2Y0A02YK0F) Distance Measuring Range: 20cm to 150cm. Two Ultrasonic Range Finder Sensor (HC SR-04) Distance Measuring Range: 2cm to 400cm.
Bread Board	Small Size Bread Board.
Communication	USB connection Serial Port.
Principal Dimensions	Height: 30cm, Length: 62cm, Width: 49cm.
Weight	Approx. 4.936kg.
Payload	Approx. 500g.
Power	Two Rechargeable Lithium Polymer 3 Cell, 11.1V, 2000mAh, 20C Battery